**ELEC 3402 Distributed Computing**

**Tutorial 1: Java client-server model**

**Student UID: 3035124881, Name: Cheng Wei**

**Submission 1: ChatServer.java code**

**import** java.io.\*;

**import** java.net.\*;

**import** java.util.\*;

**public** **class** ChatServer {

**public** **static** **void** main(String[] args) {

**new** ChatServer().go();

}

**private** ArrayList<ClientHandler> clients;

**private** **int** numOfClients;

**public** ChatServer() {

clients = **new** ArrayList<ClientHandler>();

numOfClients = 0;

}

**public** **void** go() {

**try** {

ServerSocket ss = **new** ServerSocket(10000);

System.***out***.println("Listening at 10000");

**while**(**true**) {

Socket s = ss.accept();

**int** thisClient = (++numOfClients);

ClientHandler client = **new** ClientHandler(thisClient, s);

System.***out***.println("New connection: "+thisClient);

addClient(client);

**new** Thread(client).start();

}

} **catch** (IOException e) {

e.getMessage();

e.printStackTrace();

}

}

**public** **synchronized** **void** addClient(ClientHandler client) {

clients.add(client);

}

**public** **synchronized** **void** removeClient(ClientHandler client) {

clients.remove(client);

}

**public** **synchronized** **void** broadcast(String message) {

**for**(ClientHandler client: clients) {

client.send(message);

}

}

**private** **class** ClientHandler **implements** Runnable {

**private** BufferedReader in;

**private** PrintWriter out;

**private** **int** id;

**public** ClientHandler(**int** clientID, Socket s) **throws** IOException {

id = clientID;

in = **new** BufferedReader(**new** InputStreamReader(s.getInputStream()));

out = **new** PrintWriter(s.getOutputStream());

}

**public** **void** run() {

**try** {

**while**(**true**) {

String message = in.readLine();

**if**(message == **null**)

**break**;

System.***out***.println(id+": Message '"+ message + "' received");

broadcast(id+": "+message);

}

} **catch** (IOException e) {

} **finally** {

System.***out***.println("Closed: "+id);

removeClient(**this**);

}

}

**public** **void** send(String message) {

out.println(message);

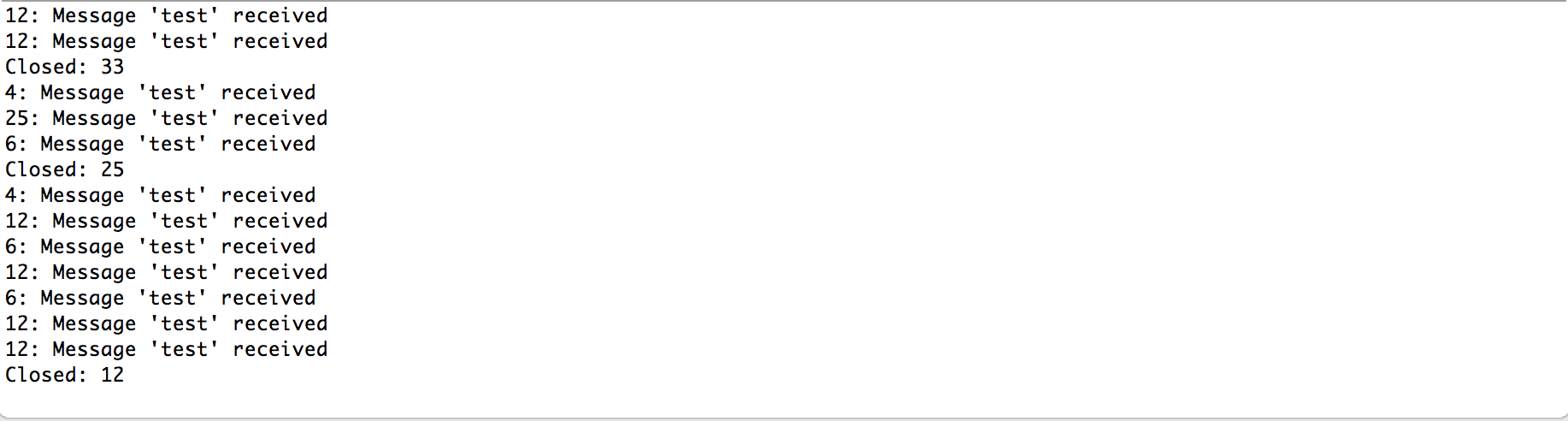
out.flush();

}

}

}

**Screenshot of client and server console:**

1. Client (ChatClientTester.java):
2. 
3. Server (ChatServer.java):
4. 

**Submission 2: Answer to the WeChat problem**

1. How much time does it take to send a Wechat text message from you to your friend?

Ans: Roughly 0.5 sec.

2. How much time does it take from uploading a picture to Wechat to others seeing it?

Ans: Depends on the image size, it takes about 1s to send a normal size image taken from cellphone.