User class

Field: userType, password, email, nameOfUser, storeName, conversationUser, blocklist, invsiblelist

Getters and setters

Functions:

isblocked(User receiver): check if parameter user in the blocklist of this user

isInvisible(User receiver): check if parameter user in the invisible list of this user

addConversation(User receiver): add parameter user to conversationUser list, allow receiver and this user make conversation

createMessage(User receiver, String message): add message between this user and receiver’s conversation log. Also add the message from receivers file to this user’s conversation log

editMessage(User receiver, String oldMessage, String time, String newMessage): replace the oldmessage at given time with new Message and time will not change. The message will be change in both the conversation file between this user and the receiver.

deleteMessage(User receiver, String time, String message): delete the specific message at given time. Only delete the message on the conversation log who initial to delete this message

display50Message(User receiver): get latest 50 messages out from the conversation log between the parameter user and this user. If there are not enough of 50 messages then just all messages.

sendTxtFile(User receiver, String fileAddress): break down the txtfile and call createMessage for how many line is in the file times.

**ALL invalid situation has checked in the program. All of that can be cut out after implementation of GUI since we don’t have to check whether the input email and name is right or not.**

**All System.out.print will be replaced to GUI based.**

NoMessageFoundException class: uh no need to explain right?

UserExistException class: uh no need to explain right?

Client class and Server Class

Server class most important field, currentUser and users(arraylist)

Server class functions:

addUser(User user): add parameter user to server user list, if user is not unique, throws a UserExistException

checkUniqueUser(String email, String nameofUser): use for create account check that if the email and name combination is unique in the server user list

authenticateUser(String email, String nameofUser): use for log in, check if the email and name combination exist in server user list

unBlockUser(User user): currentUser’s block list remove the parameter user

unInvisUser(User user): currentUser’s invisible list remove the parameter user

exactPerson(ArrayList<User> listofUser, String email, String name): find if the given email name combination exist in the parameter listofUser

searchValidUser(String searchingUsername): use to find matching username in the server userlist, like a search engine. It will NOT include user that is same type as currentUser(like seller and seller). And also it ignore upper and lower case

allVisibleStore(): use for customer, return all visible seller that exclude invisible ones(seller invisible to customer or vis versa)

currentVisibleCustomer(): use for seller, return all visible customer that exclude invisible ones(seller invisible to customer or vis versa) since it is seller side, it cannot see customer that seller haven’t have conversation with.

displayMessage(ArrayList<String[]> messages, int amount): display amount of message in the parameter message arraylist, message are displayed as “sender-time-content” format. Bottom is the latest message

There are some common blocks in client class and server class

Type 1:

A screen shot of a computer program

Description automatically generated

It just checking invalid condition

Type 2:

A computer screen shot of a black background

Description automatically generated

This is when server read from client input, split(“,”, 2) since those input will come in as (email,name)

Type 3:



This kind block just combine every element in a list and sent them to client since String is the only variable type we can send through

Since stream and lambda (Something like this (n -> asdlkfjklasdj) and stream()) is a bit hard to understand, I have commented all of them for understanding.

When reading the client and server, please reading match the line then read, otherwise it is pretty hard to know what this is actually for

Work flow (Server and client is matched)

// Welcome (wrote) (Server side will read all preivous user data)  
// Connect to server (wrote)  
// login (wrote)  
 // have or dont have account  
 // create account -> unique or not unique (wrote)  
 // login in -> can or cannot (wrote)  
// Main page  
 // display (wrote)  
 // One page of recipient (wrote)  
 // if user is seller, list of customer (wrote)  
 // if user is customer, list of stores of seller (wrote)  
 // operation/option for user  
 // add conversation button (add Conversation button -> search people that can make conversation(opposite role)(only exist users)) (wrote)  
 // check existing conversations (also button, display latest message with time and name) (wrote)  
 // account modifications (wrote)  
 // change email (wrote)  
 // change username (wrote)  
 // change password (wrote)  
 // delete account (wrote)  
 // block user (wrote)  
 // unblock user (wrote)  
 // invisible user  
 // un invisible user  
 // log off (wrote)

Missing operation, invisible user and un invisible user

Current problem:

Problem in handout that didn’t solve:

1: “All file imports must occur as a prompt to enter the file path”: I am not sure if the files like conversation logs are need to prompt to enter(when doing like create/edit/delete message). If so, general Structure need to change.

2. Invisible is not implemented, I will do it tonight

(For project 5, I haven’t solve the multithread part. Network I/O go with it so ye I haven’t solve yet. But if these 3 files all works then project 4 will be good. (I am not sure about this but I think that if we miss things in project 4 that didn’t work or didn’t implement, if we add those into there and project 5 everything works, we will get points back)

3. I am not sure about the stores thing, might need to add that in work flow that if currentUser is a seller, he can add more stores but I will check on that and if it is requirement, I will add it tonight

4. Haven’t debugged

(Sorry I wrote too much and didn’t leave it to u guys, but I really think that one person writes this kind of program the fastest and since we don’t have much time left for project 4, we all want a good grade in this class so ye)

For trading, after project 4 is submitted, I will be pretty busy because of another class I have in the summer. So GUI(I asked and complex GUI is better so ye) will be mostly for you guys, I will finish the thread asap.

Things for you guys

1. Check the user class’s function, I have tested create/delete/editMessage, but not the display50Message and sendTxtFile. Also the once in client, like take them out and create another file and test.
2. Create a testing file for check the workflow of client and server, the best will be it all work. If it run into problem, just upload the testing file to github and I will debug
3. The project 4 report(Create a google doc). For the portion of contribution(including write the report), just put something up in the program. For the what can you do different part, just say can learn lambda and stream. I will say in my part that I change you guys’s code to lambda and stream to minimize some long loop. (All the functions and workflows I have put up on the top, when writing description, can use them)
4. I think they might check on github upload to check contribution. I am not sure how will they check, but I think you guys can reupload your version of the program to there to show contribution(Again I am sorry about didn’t let you guys practice the thing learn in the class). Or reuploading might work
5. Don’t worry about peer review, it will be fine.
6. Coding style please help fix thanks.