SOFE 4790U: Distributed Systems (Fall 2023)
Instructor: Dr. Ahmed Badr

Assignment #1

Honour code: By submitting this assignment, I (name and banner id# below) affirm this is my own work, and I have not asked any of my fellow students or others for their source code or solutions to complete this assignment, and I have not offered my source code or solutions for this assignment to any of my fellow students.

Name: Liam Rea

Banner ID#: 100743012

1. Application idea

A chat app that connects you to another person either synchronously or asynchronously and allows them to chat. The app should handle multiple people chatting at the same time by using threads for each person.

2. Describe the two core functionalities

The two core functionalities of the app are the ability to login to the app using a username and password, and the ability to send messages from one person to another, either Synchronously or Asynchronously, meaning they do not have to be online to receive a message, they will get it once they login.

3. Describe the two novel features

The first novel features of this app are the ability to save messages from one person to another by selecting who you want to send to and storing the messages in a database

The second novel feature is the ability to make a notes section by selecting yourself as a recipient. Using this feature, you can take notes that will show up the next time you enter notes mode.

4. Challenges and solutions

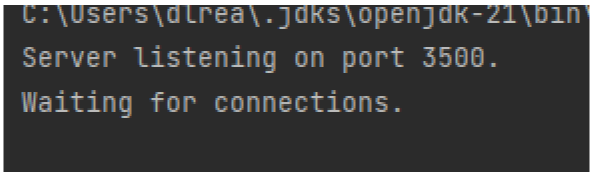
Part of the challenge I had with this assignment was getting the database set up as I had never used SQL with Java before. I installed MySQL workbench and MySQL and the driver to communicate between them. Setting up the database, I got basic SELECT and INSERT statements working which allowed me to start setting up the program.

Using the lab 1 task 4 as a basis, I encountered the issue of being able to get messages from the database actively. To solve this issue, I created another thread that can be called from one of the connection threads. This allows each thread to have essentially a sending thread and a receiving thread at the same time. Both are looking for input, one from the database, and the other from the client. The client was also adjusted, creating another thread to constantly detect if there were updates from the server and print them to the console.

Also, the output would need to be adjusted for the notes mode, requiring that the messages didn't print out to the client. As such, the secondary thread was disabled for notes mode.

5. Testing

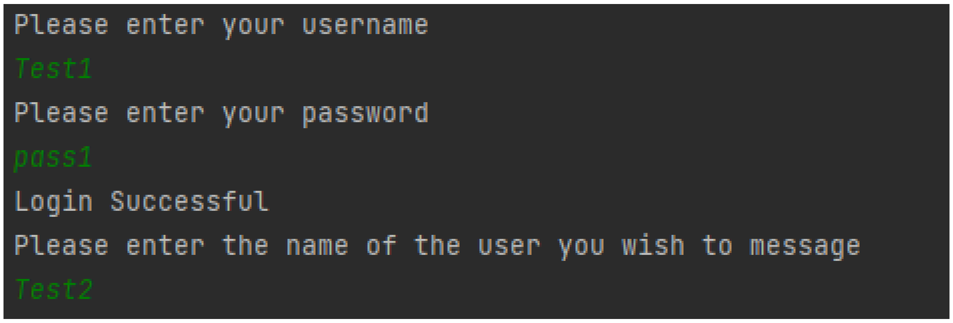
Test1: Connect to a second person and send a message, message is saved when offline, walks through logging in an sending a message to a user



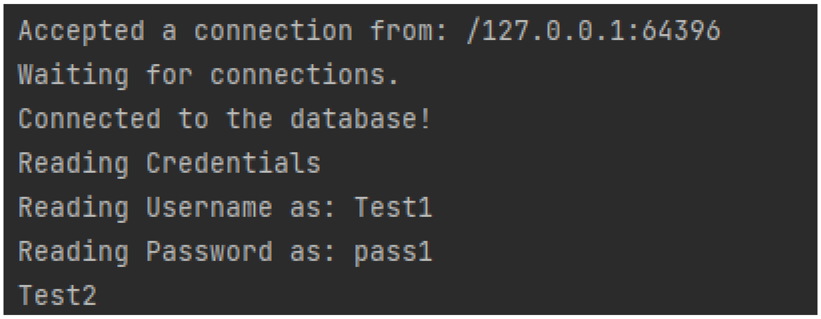
```
C:\Users\dlrea\.jdks\openjdk-21\bin\
Server listening on port 3500.
Waiting for connections.
```

Server Starts up and waits for incoming connections

Client logging in as Test1 and requests to message Test2



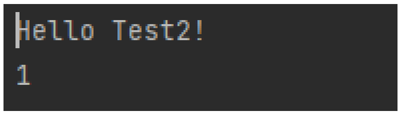
```
Please enter your username
Test1
Please enter your password
pass1
Login Successful
Please enter the name of the user you wish to message
Test2
```



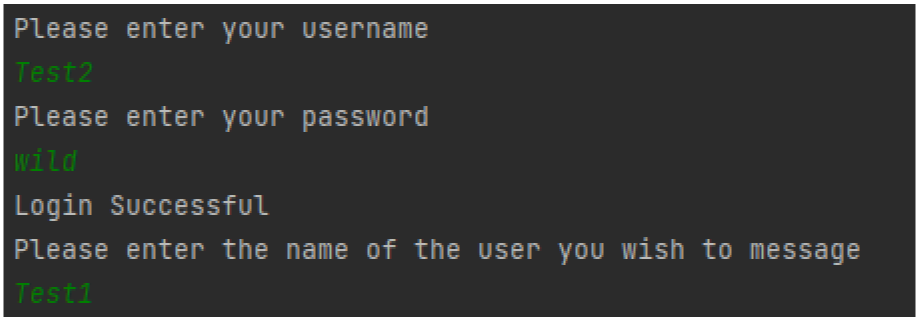
```
Accepted a connection from: /127.0.0.1:64396
Waiting for connections.
Connected to the database!
Reading Credentials
Reading Username as: Test1
Reading Password as: pass1
Test2
```

Server output log of accepting the connection and username and password. Also receives the desired user as Test2.

Write a message to test2, server receives it and logs it to database

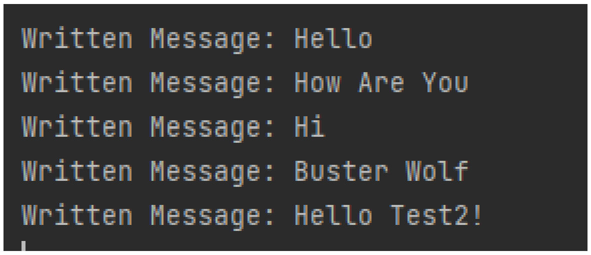


```
Hello Test2!
1
```



```
Please enter your username
Test2
Please enter your password
wild
Login Successful
Please enter the name of the user you wish to message
Test1
```

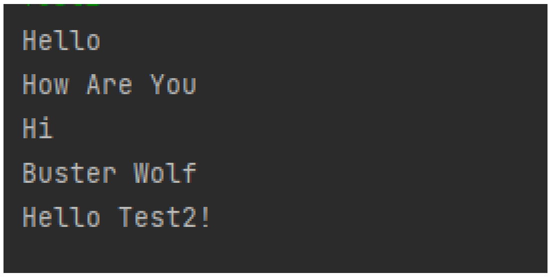
Creating another client, logging in as Test2, requesting to message Test1



```
Written Message: Hello
Written Message: How Are You
Written Message: Hi
Written Message: Buster Wolf
Written Message: Hello Test2!
```

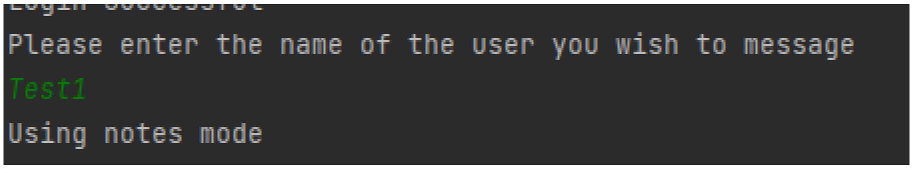
Server sends messages from database to client, these are the messages at present from Test1 to Test2

Test2 Has received messages, even when offline!



```
Hello
How Are You
Hi
Buster Wolf
Hello Test2!
```

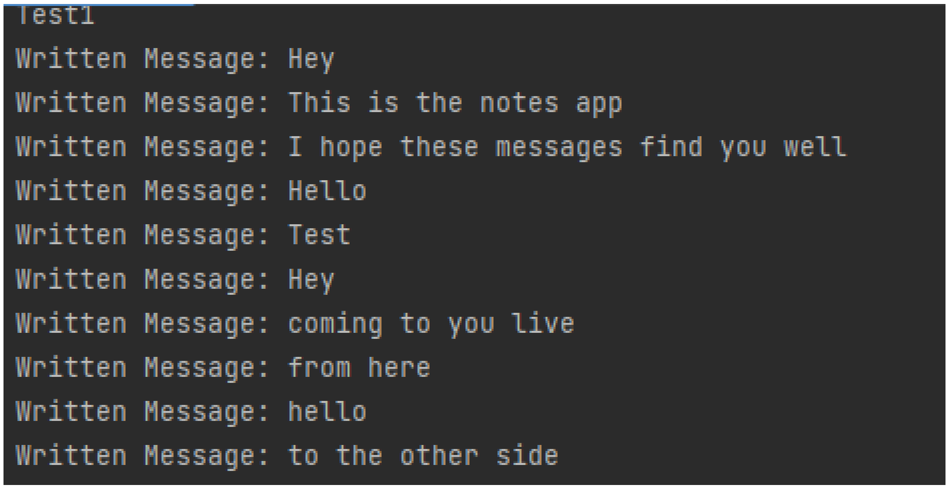
Test2: Write notes to yourself using the notes mode. Requires entering the recipient as your own username



```
Login Successful
Please enter the name of the user you wish to message
Test1
Using notes mode
```

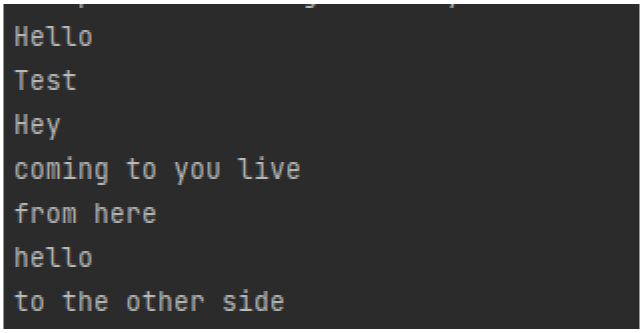
By requesting the same user as our username, we enter notes mode.

Server Connects and displays notes written by user



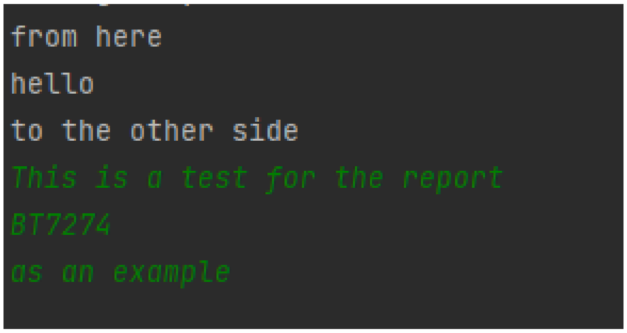
```
Test1
Written Message: Hey
Written Message: This is the notes app
Written Message: I hope these messages find you well
Written Message: Hello
Written Message: Test
Written Message: Hey
Written Message: coming to you live
Written Message: from here
Written Message: hello
Written Message: to the other side
```

Sending back to Client



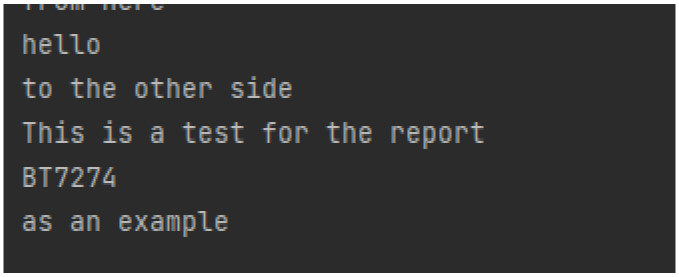
```
Hello
Test
Hey
coming to you live
from here
hello
to the other side
```

Sending Notes



```
from here
hello
to the other side
This is a test for the report
BT7274
as an example
```

Re-logging as test 1 shows that the notes were saved!



```
from here
hello
to the other side
This is a test for the report
BT7274
as an example
```