Faculty of Science

Course:	CSCI 2020U: Software Systems Development and Integration
Lab:	#10
Topic:	Multi-threaded socket application

Overview

In this lab, you'll write a simple bulletin board server and client. The server will support multiple clients by spawning a thread to handle each one. Each client will be able to post messages into the server.

***Note**: You can create a different application, if you'd like. However, what you complete must use sockets to communicate, and must use multiple threads. If your application uses a shared state between threads, be sure to protect it using synchronized or another similar mechanism.*

Instructions

You can use any operating system or environment for this laboratory assignment.

You will create a new directory (or IntelliJ IDEA project) called `lab10`.

The Client

Write a simple bulletin board client, with the following user interface elements:

- Username – a text field where the user can enter their preferred username
- Message – a text field where the user can type messages
- Send – a button that, when clicked, will send the message to the server
    - Note: Don't forget to flush() the socket's output stream when sending the message to prevent buffering, since we want this to be interactive
- Exit – a button that, when clicked, will close down all connections and exit

The client does not need to have multiple threads.

Messages being sent will use the following form:

```
Username: Message text
```

***Note**: This bulletin board will use the honour system. If the user changes their username, that is just fine. You will simply use whatever username they enter in the message.*

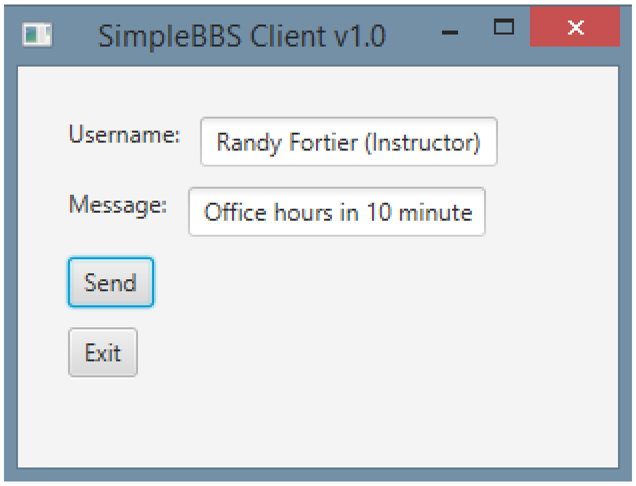


Figure 1 – The client window

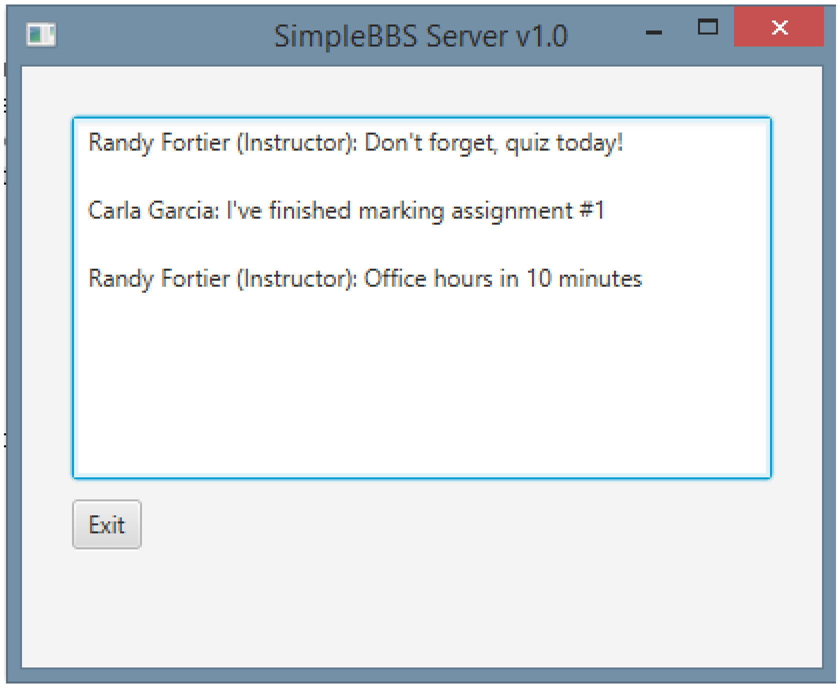
The Server

Write a simple bulletin board server, with the following user interface elements:

- Messages – a text area showing all of the historical messages
- Exit – a button that, when clicked, will close down all connections and exit

Use the multi-threaded web server as a template. Every time an incoming connection is established, spawn a new thread to deal with that client. When messages come in, they are added to the messages display.

***Note**: It is not required, but recommended, that you write another thread class to handle the listening of incoming connections, so that your main thread can be used exclusively for the user interface (e.g. button presses).*



```
Randy Fortier (Instructor): Don't forget, quiz today!

Carla Garcia: I've finished marking assignment #1

Randy Fortier (Instructor): Office hours in 10 minutes
```

Figure 2: The server window

How to Submit

Show your running application to the TA to prove that you have finished this lab. The TA can provide oral feedback if you do not receive full marks for any lab assignment, but it is most appropriate to ask the TA for this feedback in a timely fashion (i.e. ask now, not at the end of the term).