Nik Srinivas
ns29374
11/30/20

Final Project 422C Breakdown

Programmers POV

Program Summary

This program is built to mimic the basic functionality of a website such as eBay. It consists of 3 main parts, the bidding client, the login client extension and the server. The users guide following this section will read more like a traditional README with instructions on how to run this program on your personal computer. This section will go part by part and discuss how I overcame/addressed certain challenges and requirements for the project.

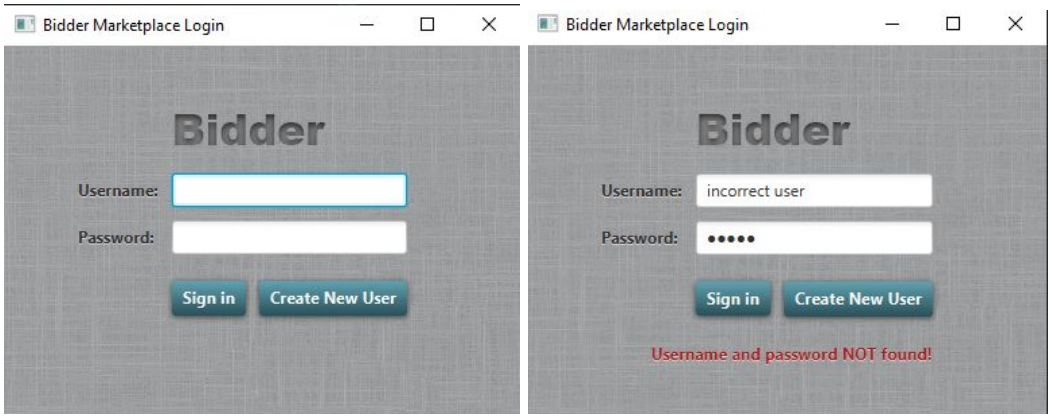
Login

There are two main types of visualization tools I used - both are sort of branches of JavaFX. There is the JavaFX library itself used for creating the Login GUI and the java swing libraries used for the Bidding Client GUI.

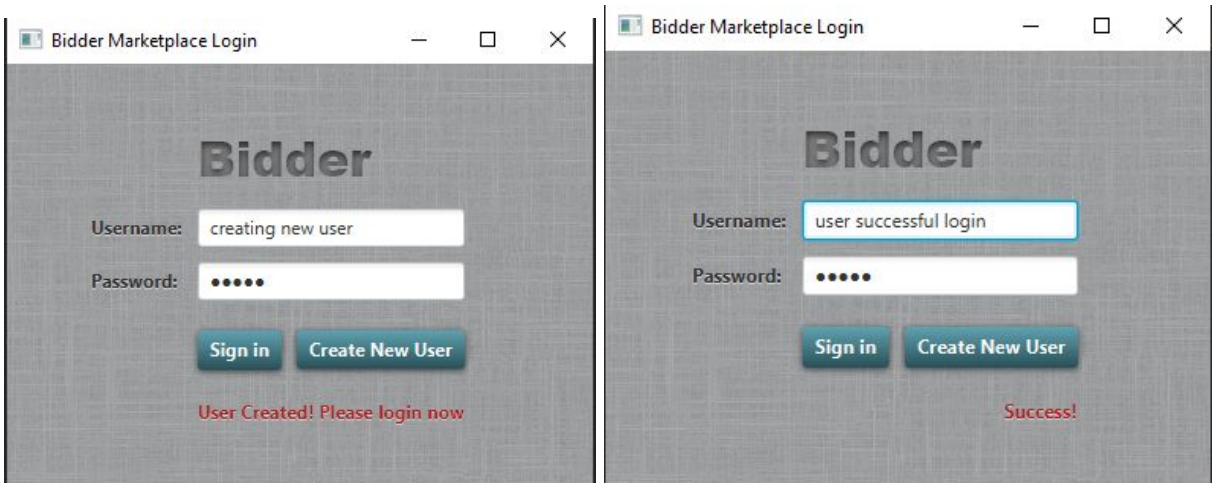
The first step was creating a GUI. I modeled mine based of the documentation found here -> https://docs.oracle.com/javafx/2/get_started/form.htm
There are 3 events to possibly handle when a user is logging in.

1) New user creating an account
2) Current User typing an incorrect password
3) Successful login

Because this logic is handled in the backend, even the logic class needed to have the socket setup to correctly network to the server. The user's password is secured by both hiding the character while typing as well as using base64 encryption when sending it to the backend. The password is saved in its encrypted state so even if the server got "broken into", the hackers still wouldn't have the correct input into the login client. Each of these 3 cases has feedback from the server that is shown on the client under the login buttons.



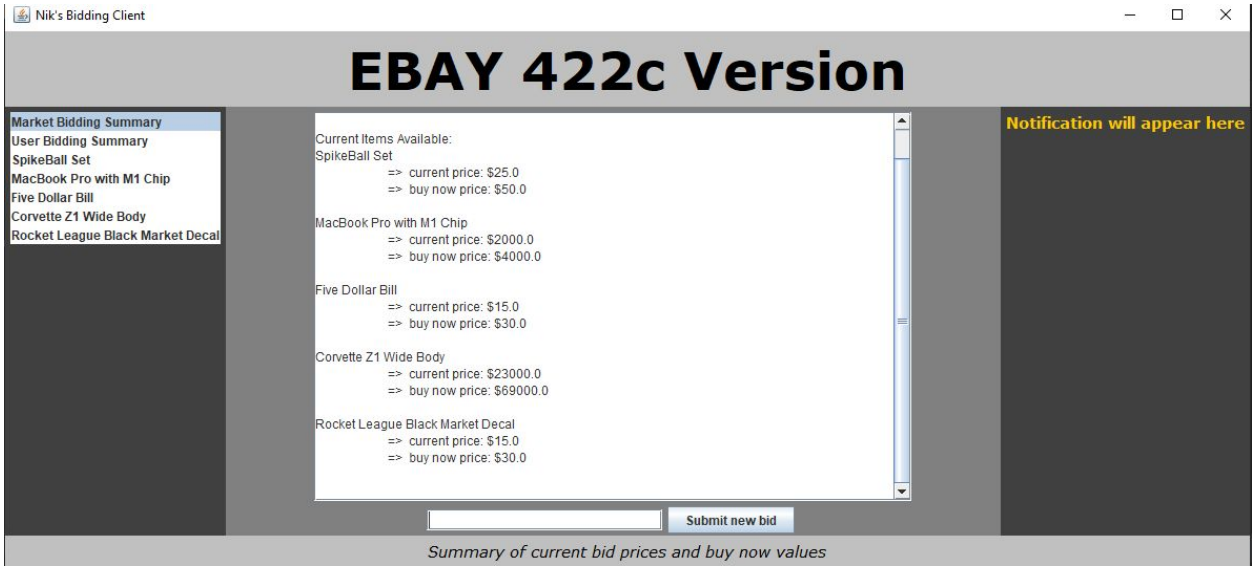
Nik Srinivas
ns29374
11/30/20



These users and their passwords are stored server side, not on the cloud, so new users must be created every time the server spins up.

Bidding Client

The bidding client GUI started from the chat room example we worked on in class and then slowly shifted to include other parts. It is based on a borderpane framework so it has 5 regions, top, left, right, bottom and center.



The left side is a list where changing the selection affects what you see in the middle. The middle is a text area that houses the prices of the items or the user/item bidding history depending on what you select. On the right side is a notification center where the most recent bid on any item is instantly shared throughout all clients and a small notification sound is played. At the bottom there is a description of each item that changes with the selection on the right side.

Nik Srinivas
ns29374
11/30/20

Because this is a text based bidding system, I thought it didn't really make sense to add a timer as it would be confusing to switch tabs to bid and not see all items at once. Instead, the stopping condition is when the user reaches a buy now price or it is the highest bidder when the server is closed.

Here is where I got the code to play sound in a JavaFX project, I had to modify it with try/catch to not throw an error with multiple threads running it for some reason.
https://stackoverflow.com/questions/26775260/javafx-audio-stopped-after-seconds

The following are smaller stylistic things that appear when initializing the GUIs

https://stackoverflow.com/questions/20767290/how-do-i-change-the-font-size-of-jpanel
https://stackoverflow.com/questions/58788684/having-a-problem-in-changing-font-in-java
https://www.daniweb.com/programming/software-development/threads/423941/how-do-i-make-jlabel-bold-and-italic-at-the-same-time
https://www.educba.com/jpanel-in-java/
https://www.geeksforgeeks.org/java-swing-jlist-with-examples/
https://stackoverflow.com/questions/20304933/how-to-get-javafx-nodes-textarea-textfield-to-resize-correctly-when-user-drag

Server

The backend is pretty straightforward since it doesn't use a cloud server. I essentially made a database with strings where the message was indexed such as

-item flag - user flag - server message

When communicating back and forth. This made it easy to key into which messages were for logging in vs actually bidding on items. This also helped ID each client to know when to send certain messages just to them (error messages) or to everyone (bidding updates).

Users POV

These are the instructions for running the program

1) Unzip the client and server files found https://github.com/EE422C/fall-2020-final-project-Nik-Srinivas
2) First we have to run the server locally so locate the file location and traverse there from your command prompt

Nik Srinivas
ns29374
11/30/20

   a) EXAMPLE
      i) cd C:\Users\ibudg\IdeaProjects\Server\out\artifacts\Server_jar
      ii) Java -jar Server.jar
3) Next double click Client.jar in the other folder to start an instance of the client
   a) Note: You must double click the client jar for each new user you want to run simultaneously

There are 6 tabs

1) Bidding Summary - gives you up to date view of all prices
2) User Summary - history of all your bids
3) 3-7 are all the bidding history of the other items

To make a bid, simply click on the tab and enter a price. New bids made by other users will appear on the notification panel on the right side of the client. If you are the highest bidder when the server closes or if you reach the buy it now price then you will get that item.

Enjoy!