NYU | TANDON SCHOOL OF ENGINEERING

CS-UY 3913 Final Project

Project Description

Sanidhya Sitaula (ss12081)
Monique Aboumrad (mas1790)

Professor Daniel Katz-Braunschweig

Introduction

This project was built using Java Maven.

1. Project Description

The application has five main components:

User Login/Sign Up

The login/sign up functionality allows for an unregistered user to sign up, and after a successful sign up, log in to access the application. The user is then redirected to the home page where he/she can view all the posts made by other users, like/unlike posts, along with other functionalities mentioned below.

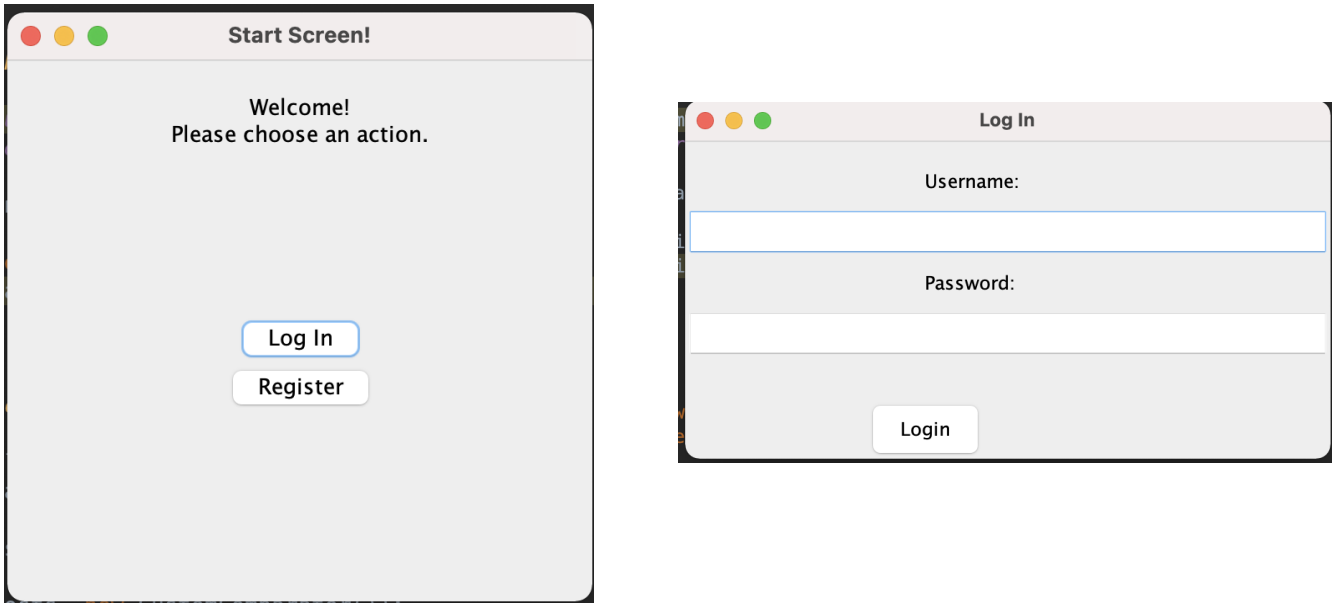

Figure 1: Screenshots of Initial Screen and Log In Screen

Technical Description

The login/sign up functionality was implemented using JDBC and H2 database. After many failed attempts at installing the SQL driver (we figured it was a Mac issue), we went with H2 since it provides the exact same functionality and has the same query structures as SQL. After signup, user data is pushed to the database. The data is then pulled after a login attempt.

Drawing Functionality

The heart of the application is having the ability to draw, using Java Graphics, and post it to the application for other users to view. The user is presented with a window with a dark canvas, and he/she has the ability to choose between different colors to draw with. A "clear" functionality is also presented so that the user can clear the screen and re-draw again.

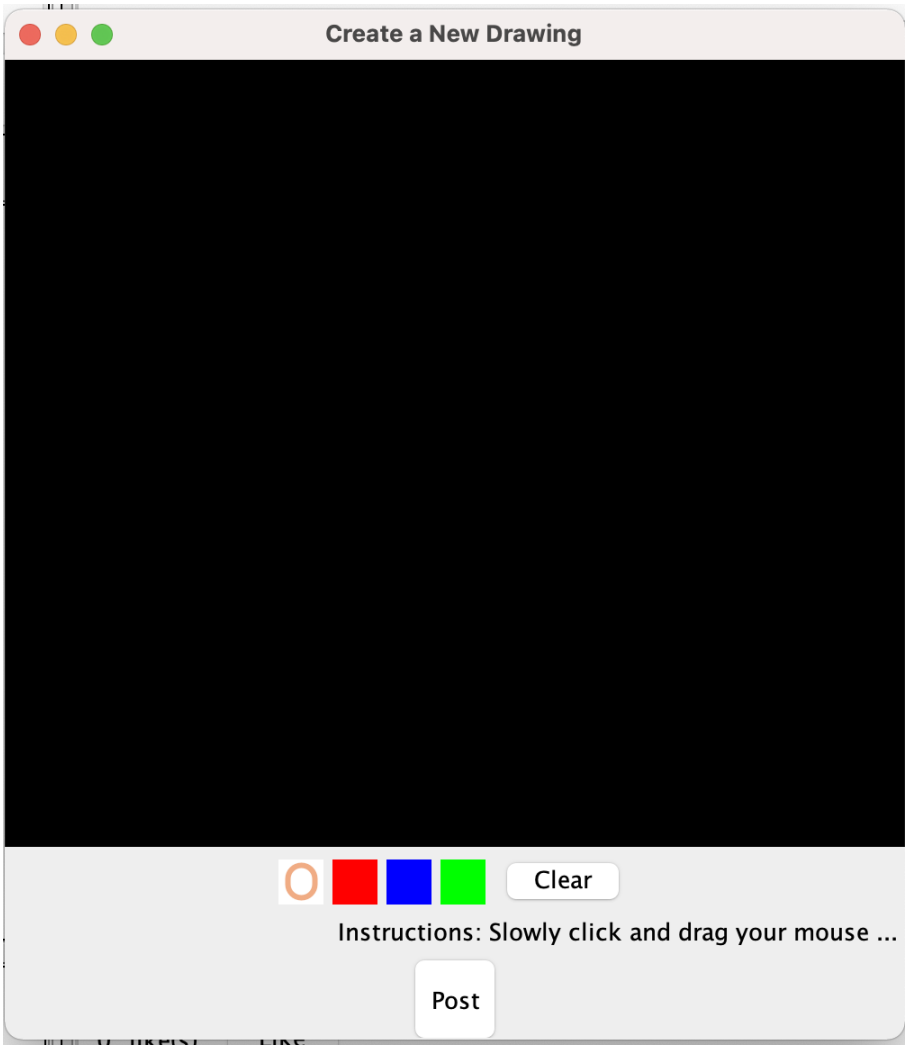

Figure 2: Screenshot of the drawing screen

Technical Description

The chosen approach for this functionality was to create two Graphics objects, one solely for displaying the current status of the graphics object to the user, and the other to keep track of the drawing so far and eventually to store as a BLOB file to the database. Mouse Action Events were chosen to listen for mouse drag activity. A combined class, named PointColor, was also created,

in order to store each point and its associated color (so that when the canvas re-paints itself, and the user selects a different color, the color information isn't lost). When the user clicks on "Post," the graphics object is converted to a `byte[]` object and pushed to the database.

Home Page

The home page consists of posts from all users registered in the application. The user then has the ability to like a specific drawing/post, and also revert the like (unlike) it if they wish to. The user can also directly access a user's profile through their post, and also access their own profiles through a panel on the left, as shown on figure 3.

Figure 3: Screenshot of the Home Page

Technical Description

The application retrieves all of the posts from the database and simply displays them neatly. Each post container has a like button and a view profile button (if the post isn't made by the user logged in). The `byte[]` object obtained from the database is converted to an ImageIcon and added to the `JPanel` object associated with each post.

Chat

Users are also able to chat with other users currently online in the system. The chat window is presented in the homepage itself in the left panel.

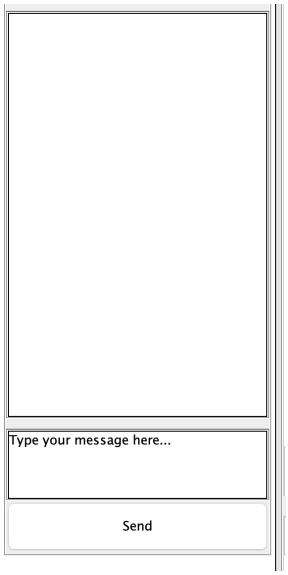

Figure 4: Chat window (incorporated within the homepage)

Technical Description

The chat functionality is implemented using Sockets. To access this functionality locally, simply running two instances of the application and logging in as different users is enough.

Profile

The user can view the profiles of other users as well as their own. The profile window shows the user's information (number of posts created, number of likes received), and all the posts made by the user, as shown on figure 5.

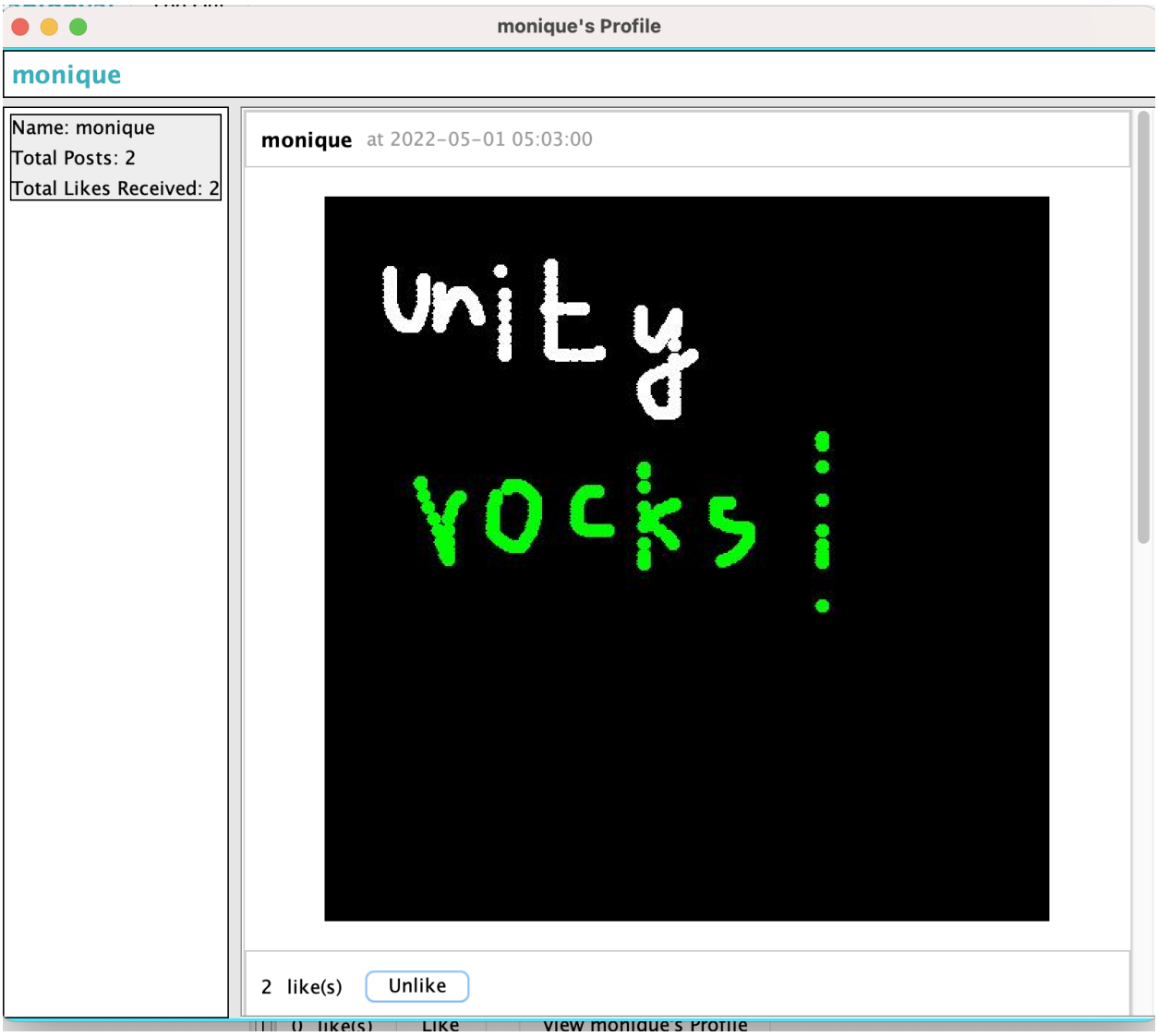

Figure 5: User Profile View

Technical Description

The profile window is loaded when the application finishes receiving the user data from the database. As shown in Section 2 of this document, each post has a “number of likes” column that updates whenever the user likes a post.

Database Description

The application makes use of three tables:

USERS

The users table the following columns:

- `id (int)`
- `name (varchar)`
- `username (varchar)`
- `password (varchar)`

POSTS

The posts table has the following columns:

- `id (int)`
- `username (varchar)`
- `date_posted (varchar)`
- `photo (BLOB)`

POSTS_LIKES

The posts_likes table has the following columns:

- `postid (int)`
- `likedbyuserid (int)`
    - This is to keep track of every user id that has liked a specific post (in order to make the unlike functionality work and also not lose data when a user logs out)