**COMP 3203 FINAL PROJECT**

**PEER-ASSISTED SIMULTANEOUS SKETCHING (PASS)**

**Jess Johnson**

**Jairus Bali**

**Oluwadara Elebute**

**17/12/2015**

1. **INTRODUCTION**

Computer network software relies on the client-server model. The client retrieves a particular service from the server through the network for a purpose. Peer Assisted Simultaneous Sketching (PASS) is a drawing and sketching computer network software tool that is a replication of Google Drawing [1]. The project came about while considering improvements or extensions that could be added to the Google Drive [2] package. Google Drive is a network software that enables users (clients) to save files on Google Cloud and access such files from any device anywhere using the principle of cloud computing. Cloud computing is a concept whereby files are stored and accessed from a web (network) of remote servers known as a cloud on the internet instead of the computer’s hard drive. Google Drive allows users to store files on their cloud infrastructure (server), retrieve files and share files with other users (clients).

Google Drive supports a wide range of file types including text files, images, videos, and diagrams. It was initially assumed that simultaneous sketching was not supported however, with in depth research it was discovered that Google Drawing was part of the Google Drive package which allows multiple users to sketch in real time; this changed the original plan of creating a drawing extension to the Google Drive package to creating a replication. Simultaneous sketching is important because several research results have concluded that some ideas are better explained using diagrams and sketches than any other methods thus increasing user’s efficiency. PASS is aimed majorly as a study tool for students to assist in group work in such a way that it is easy to communicate ideas; this is where the ‘Peer Assisted’ originates from.

The goal of Peer Assisted Simultaneous Sketching is to recreate the simultaneous sketching aspect of Google Drive. The main objective of PASS is to make the sketching experience as efficient and convenient as possible. PASS should allow single and multiple people to sketch in a session after creating an account or logging in with the right username and password. In the case of multiple user sketches, the sketches have to be real time i.e. the other users have to see the sketch at the same time another user is sketching. Another feature of PASS is to have the sketches look as smooth as possible with no observable breaks in lines.

PASS is a success according to the goals highlighted. We were able to create a real time simultaneous sketch software although not as complex as Google Drawing. The finished project allows single and multiple users to sketch, undisrupted by line breaks, stores users on the server and allows users to change the sketch colors. The user’s sketchpad is the client. Each time the coordinates of the line being drawn changes, the client sends a request to the server to update the coordinates on all clients in the same session.

The upcoming sections include the following. Section 2 contains the background information on the overall structure of PASS including the libraries used to implement PASS. Section 3 divides the obtained result into sections and describes how that result was obtained; it also includes the software interface. The evaluation plan follows in Section 4 and the results of evaluation in section 5. We conclude with Section 6. An appendix is also included that shows the questionnaire used for evaluation.

1. **BACKGROUND**

**2.1 *OVERVIEW*:** Peer Assisted Simulation Sketching (PASS)

PASS is a networked application allowing a team of people to facilitate the conveying of their ideas with one another. A team will have a single virtual sketch which is shared with other remote users, and each change made to the sketch will be seen by the other team members.

PASS uses a client-server architecture model where a client directly communicates with the server and then the server passes on the client's message to the other clients i.e. updating them. This updating behaviour is applied to every client modification to the sketch. The application utilizes the Model View Controller (MVC) architecture. Encapsulating the application's components into three main parts:

* Model: The model is within the client application which manages how data is to be made i.e. creation of lines
* View: The sketching window where drawings are viewed
* Controller: The server interprets every actions sent to it from the remote users and updates the clients i.e. it coordinates the model and the view

The program is entirely written in the JAVA programming language along with open source libraries. JAVA is the most appropriate language for the PASS project because of the available, rich standard libraries which makes it convenient to use. JAVA is also the programming language that PASS team members are most familiar with and comfortable writing in.

**2.2 *LIBRARIES*:**

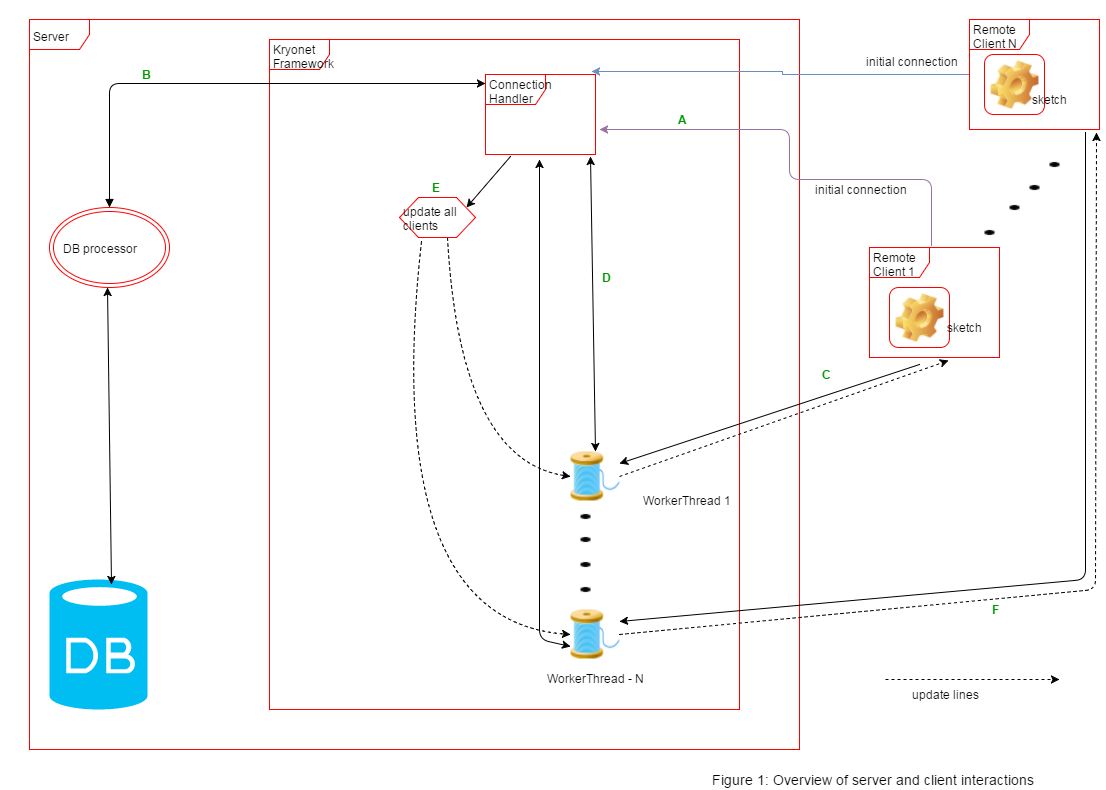
Open source libraries used are

* Processing
* SQLite with Java Database Connectivity (JDBC)
* Kryonet

Processing is used in creating the heart of the interactive experience used in PASS. It creates the sketching window and grants the user the ability to draw on it. Processing is designed to be a language driven by visual context for developers learning their first programming language [3]. However, this does not make the library any less useful to developers with a few tools under their belt already. Since the Processing library is built on top of JAVA and it compiles into pure JAVA, it was an easy fit into our project. With Processing’s heavy focus on letting visuals act as a language guide, this open source library was also a perfect fit for PASS’s goal of allowing users to let visuals, i.e. drawings, better communicate their ideas.

SQLite is a lightweight Structured Query Language (SQL) database engine which directly reads and writes on the file system on which it is deployed onto. This is used in keeping track of registered users and validates users who try to login to the system. JDBC is a database Application Programming Interface (API) that comes prepackaged with the Java Development Kit or JDK for short. The JDK is a requirement for running and developing JAVA programs, and since JAVA is cross-platform, using JDBC would allow users to have registered accounts without needing to install a separate program to manage those [4]. One of PASS’s goals was to have our drawing application be as accessible as Google Drawing is, and using SQLite with JDBC allows for that.

Kryonet is the main networking engine for PASS. It handles new user connections, messages and disconnections. The Kryonet library uses an efficient serialization library to ensure that sending objects across a network is as quick and reliable as possible [5]. Since Kryonet is a JAVA library it is useable from many platforms, both desktop and mobile i.e. Android. Kryonet has the capability of sending messages both via Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). However, due to the essential requirement of PASS that the users be updated at every stroke a remote user applies on the sketch canvas, TCP is the main networking protocol used. Using a UDP connection might create some reliability issues, meaning that remote users could potentially have incomplete drawings and thus incomplete ideas would be shared. Figure 2.1 shows an overview of how connections and messages are handled with Kryonet.

**Figure 2.1**: Overview of server and client interactions in PASS

Following the path in Figure 2.1 of the interaction of Remote Client 1 (assuming a happy path)

1. Remote client connects to the server via the Connection Handler and is given a designated worker thread to be the primary liaison between the server and the client. \*\*client and server connection is fully established\*\*

* In the client side - the sketchpad is created in the background but it is not visible

1. The server then validates that they are registered within the database

* Once validated, the Connection Handler tells the client application, via its worker thread, to show/set visible the Sketchpad interface for the client to draw on
* If it is not a valid user login the server requests that the client application terminate, and then the server disconnects with the client

1. Client modifies the sketch from their local machine
2. Those changes are then sent as SketchLine objects, sent back to the Connection Handler via Worker Thread
3. That SketchLine object is then echoed to all the connected clients to update their sketch view
4. All clients are updated except for the one that has made the change – as it would just be redundant

A good illustration to explain this is a restaurant. The client is a guest at a restaurant. The first person a guest meets at the restaurant is the greeter (connection handler) that says “hello”. The greeter then takes the guest’s name and checks the list of tables that are currently reserved (server must validate registered user in database). If the guest is already reserved for a table, then the greeter will immediately lead the guest to their table (user has a registered account and can sign in). Otherwise, the greeter writes the guests name in the reservations list and once it is the guests turn to be seated the greeter leads the guest or guests to their table (the user must first register an account and sign in). Once this is done, the greeter informs a waiter of the guests’ presence and the guests never see the greeter after that. All other interaction is done through the waiter (worker thread).

1. **RESULT**

**WHAT HAS BEEN ACCOMPLISHED?**

PASS accomplished these milestones

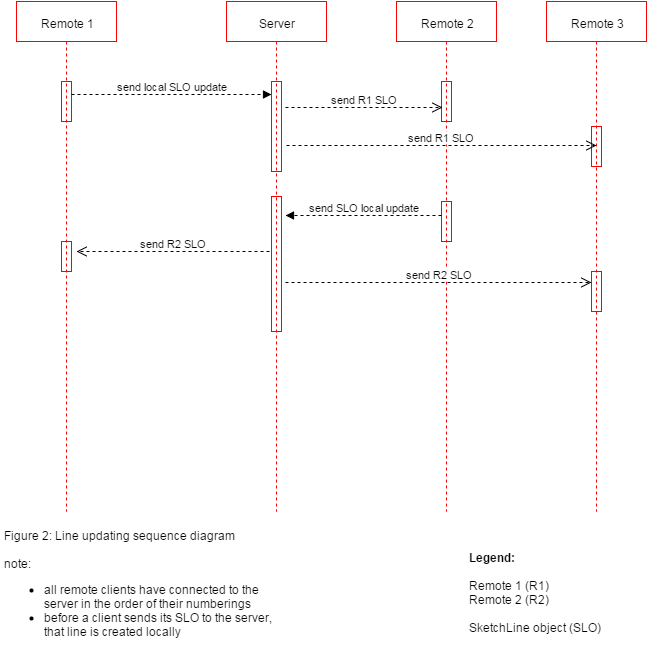
* Logging in users
* Single user / multiple user parallel simultaneous sketching
* Changing colours

**3.1 *Logging in users:***

Before a client can join a sketching session s/he must be registered to the PASS system's database. Only then will they be able to collaborate with others remotely. SQLite is the database management system used to keep track of registration and logging in of users. There is a JDBC SQLite operations processor on the server for this function which returns a message of successful or invalid log in before a client’s sketchpad can be visible.

**3.2 *Simultaneous Sketching:***

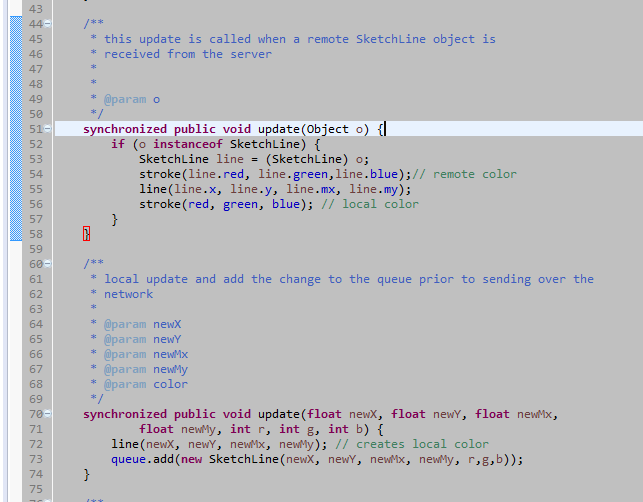
The main goal of the project PASS has been accomplished. That is that sketches can simultaneously be viewed and modified from different remote machines. Using the TCP networking protocol, each remote machine's change to the sketch is seen by everyone. Since **every** SketchLine object, i.e. modifications, has to be sent over the network to reflect a change to the sketch, all of them were collected in a queue prior to sending them. Due to the thread switches within the client applications, without this queue there were data missing when sending updates to the server. This queue was required in order to create a consistent sketch for all connected machines. To be specific the data structure used was the java Concurrent Linked Queue. This is a queue which is thread safe and it guarantees that all data is sent in a first-in-first-out (FIFO) manner.



**Figure 3.1**: This is a sequence diagram of how the line updating between the remote sketches are updated. As shown all of the SLO updates received by the server are echoed to the clients in the order of when the clients connected with the server. Even though the application may appear that local line drawings and updates from remote clients are executed simultaneous they are in fact executed in a sequential/interleaved fashion.

**3.3 *Color:***

By pressing the 'c' button on the keyboard, a color picker interface [6] will become visible for the user to modify their color line. The color is a property of the SketchLine object, using the Red Green Blue (RGB) color model. When a SketchLine object is sent over the network the color property is sent along with it.



**Figure 3.2**: Client application – line/color updating

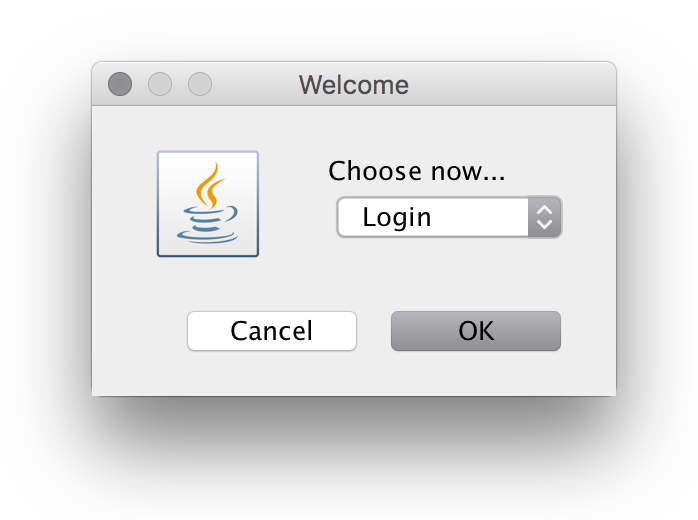
Line 51-58 in figure 3.2 shows the method for which the client application updates the local view **after receiving a SketchLine object from the other remote users indirectly from the server**. The following steps are then executed in sequence to maintain line and color integrity:

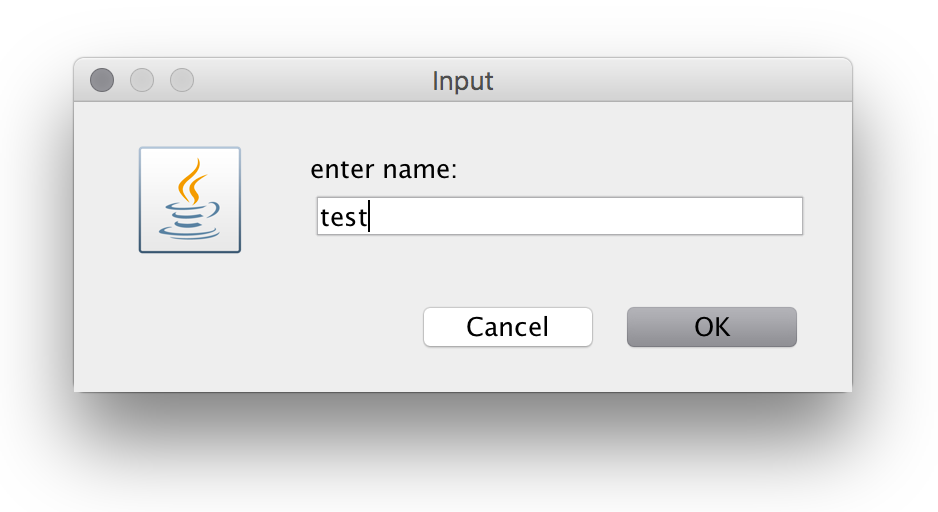
1. Line 54 -- Set the stroke color using the derived colors from the received SketchLine object. By calling the stroke(int red, int green, int blue) function, a built-in Processing function, subsequent lines drawn will be that of the passed-in RGB parameters
2. Line 55 -- Draw the received line
3. Line 56 -- Set the stroke color back to that of the local user's chosen color

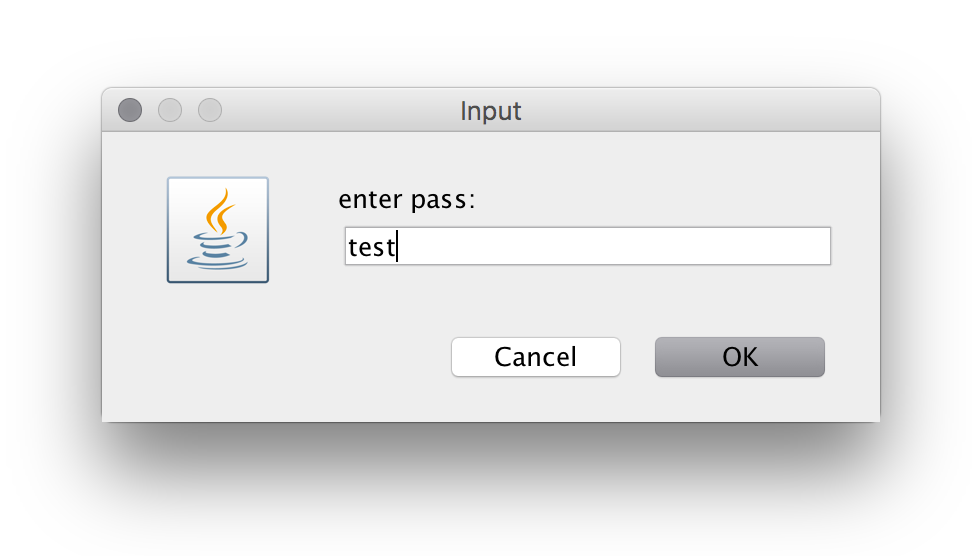
Line 70-74: The method for which the client uses to update their **local** view. As is seen on line 72 a line is drawn using primitive data. Subsequently in line 73, these primitive data are wrapped into SketchLine object and are put in a queue prior to sending it over the network, to tell the server of the changes made to the local sketchpad.

A point to note of is that the methods are **synchronized** which limits only one of the functions to be executed at a time. Due to the many threads accessing SketchPadApplet (the processing core) this keyword is necessary to uphold the integrity of updating the lines and its color.

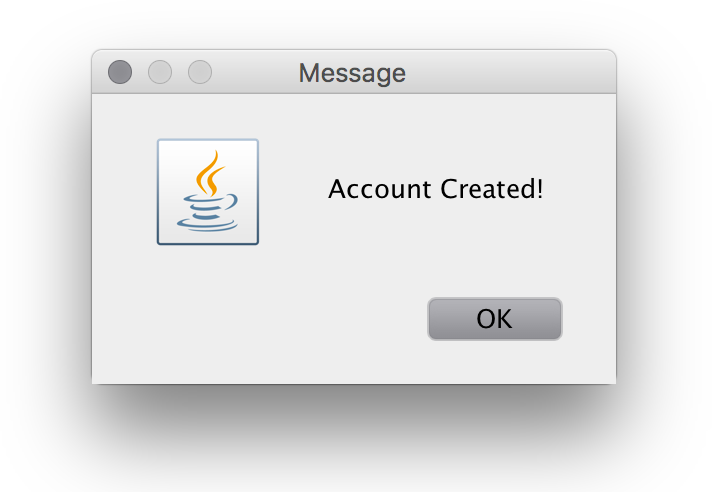
**3.4 *Software Interface*:**



**Figure 3.3.1**: The login dialogue box - When a user first connects to PASS, they are prompted to either login or register

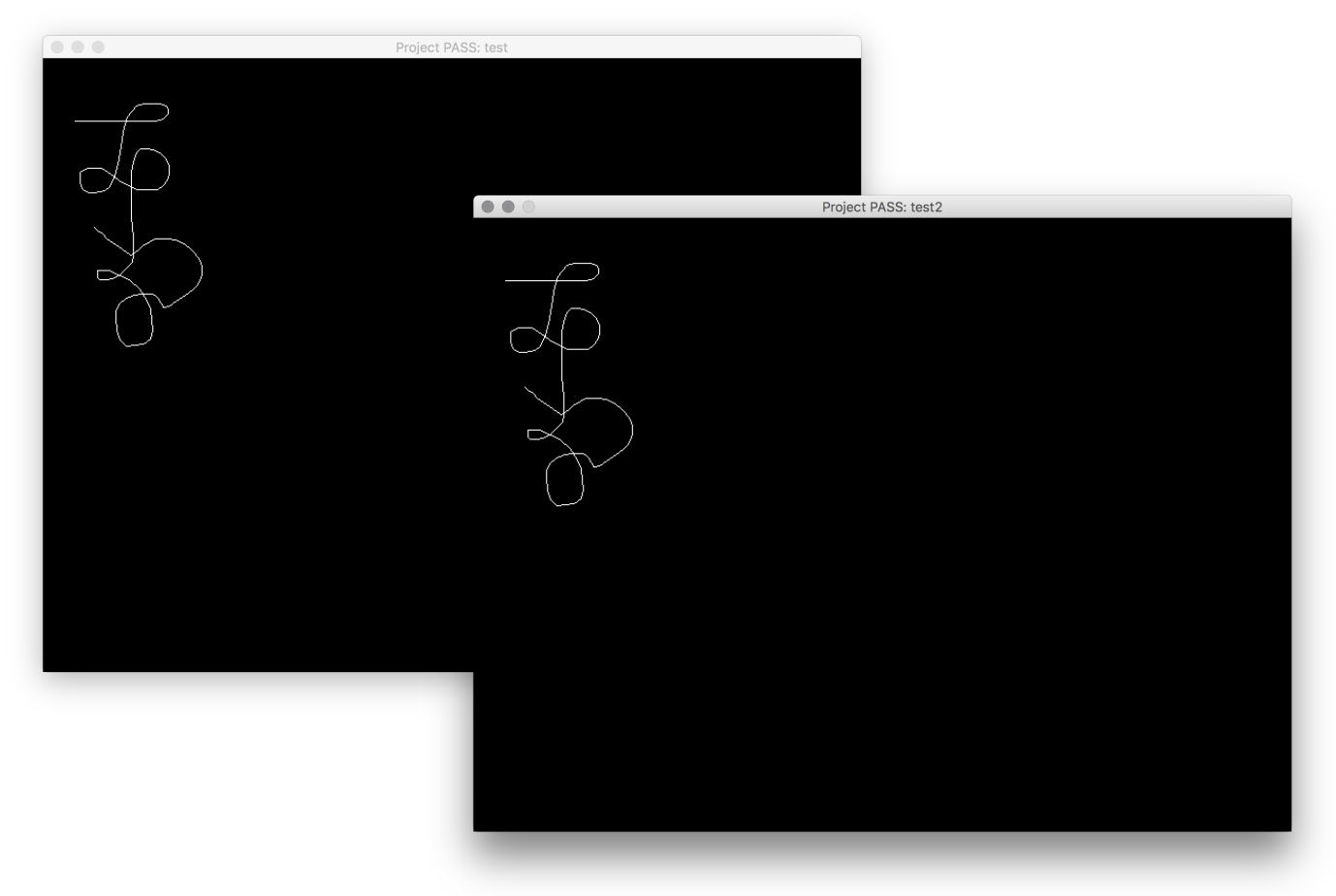


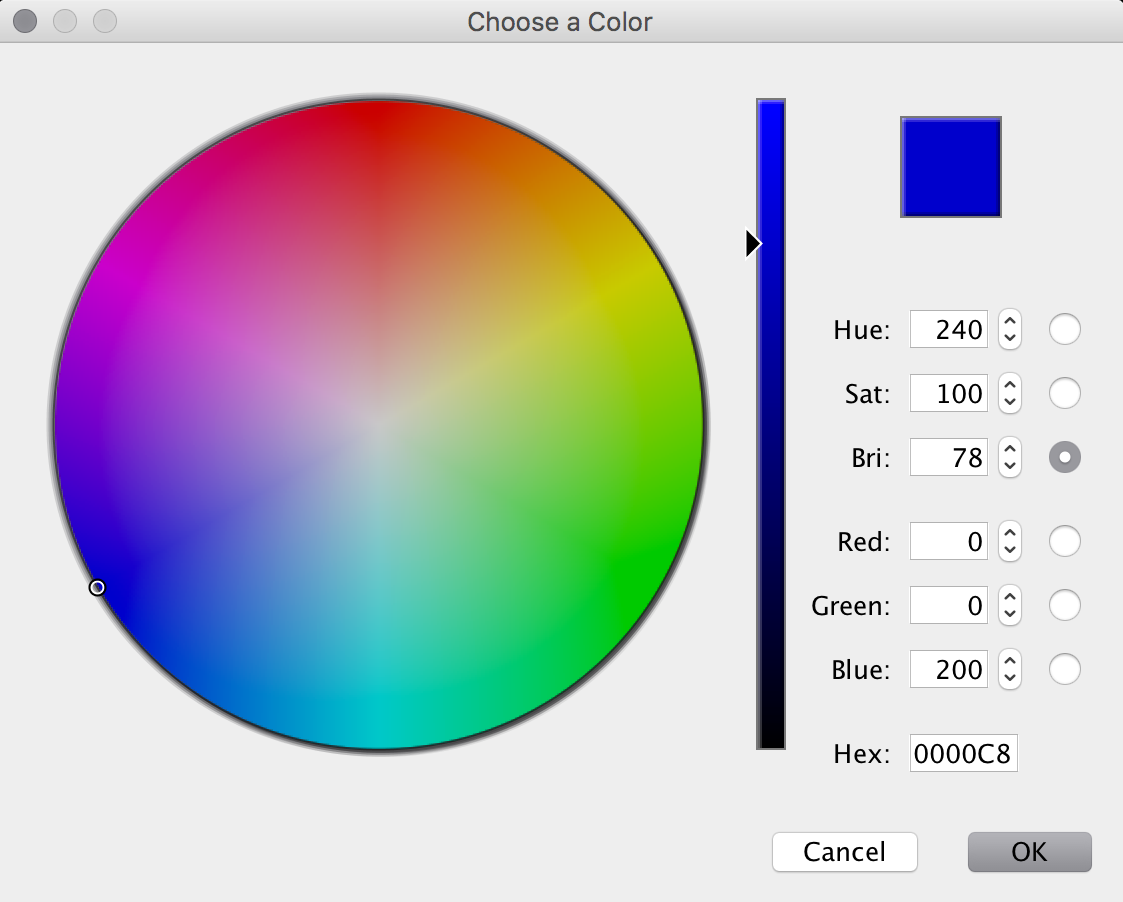
**Figure 3.3.2**: If the user is already registered, they can select the login option from the dialogue and then proceed to enter their username and password. Otherwise the user should register an account.



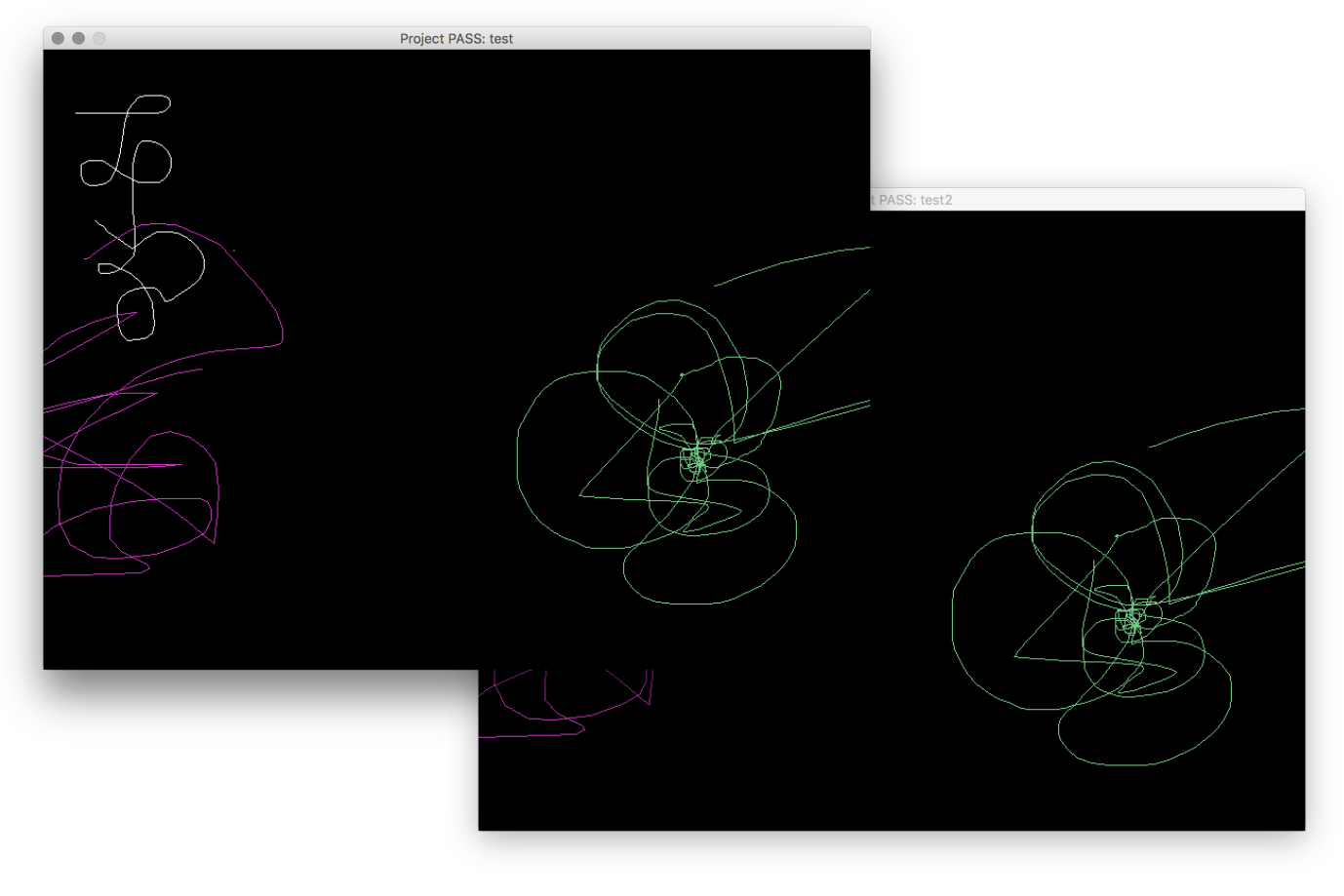
**Figure 3.3.3**: When the user has registered, they will see a dialogue confirming that they have registered and will then be redirected to the sketch canvas to begin collaborative drawing.

**Figure 3.3.4**: Shows a single and double user case, if the user has successfully logged in by entering the matching username and password combination stored in the PASS database, they will be redirected to the sketch canvas to begin collaborative drawing.

**Figure 3.3.5**: The username of the current user is shown at the top of the sketch canvas. The users may now begin drawing simultaneously. (double user case)



**Figure 3.3.6**: Shows the color palette. Users have the options to change what color of line they would like, in order to better differentiate between multiple users.

**Figure 3.3.7**: A triple user case. Although the third sketchpad is not seen here, user 1 draws in white, user 2 draws in pink and user 3 draws in green. This shows how users can be differentiated in a multiple user case.

**3.4 *How much can collaborate at a time***

The prototype of PASS does not currently implement a maximum amount of remote users to connect to a single sketching session. Therefore the main bottleneck on how responsive a sketching session is consists of: how quickly the remote users and server can communicate with one another over a network, and how quickly the server can process all client requests. As will be seen in the evaluation, the reduction in responsiveness with an increase of users was compared visually.

1. **EVALUATION PLAN**

The main goals of PASS focus on its efficiency, convenience, smoothness in each stroke of a sketch and ability to update real time on each users sketch pad.

PASS will be evaluated with single and multiple users. We use non-computer science students in study groups who are studying for an exam and use PASS as a study tool. These students are actually using it towards a specific purpose and in a tensed environment they are more time conscious therefore they will be able to assess the performance better. Nonetheless, we also use some regular students without exams. These students should have some form of previous experience with Google Drawing.

The evaluation Process takes place as follows

1. Users will be given a brief description of what functions PASS is to accomplish and how to use it.
2. After the brief introduction, the users will then be asked of their expectations of PASS; this will help to judge their evaluation results.
3. After using PASS for a given period of time, the users will be given a Likert scale of 1 to 5 to rate PASS based on each highlighted feature:

* Smoothness
* Responsiveness in updating on every user
* Efficiency/Ease of use
* Overall usability

1. Users will then be given a questionnaire to evaluate the similarities and differences between PASS and Google Drawing based on the highlighted features mentioned above, what they liked best about PASS and what could be improved

A simple visual test will also be done to evaluate how the responsiveness relates to the number of users. This project is a success if a minimum of 80% of users conclude that the highlighted features in PASS are similar to Google Drawing and the responsiveness does not decrease drastically.

1. **EVALUATION RESULTS**

**5.1 *PASS results without comparison***

A total of 8 study groups and 5 additional individuals (a total of 30 students) were given the questionnaire in appendix 1. When asked about their opinions prior to using PASS, all users agreed that PASS sounded like a useful study tool based on the introduction. The result of the questionnaire (appendix 2) after testing PASS is shown below in figure 5.1. As is shown below, the highest rating for PASS is its smoothness and responsiveness with 100% above average rating. The lowest rating was the efficiency and ease of use. Users suggested that including additional tools to PASS will make ease of use better. For example, one user in a study group needed to draw a table which was hard to do with scribble lines therefore a ruler tool in this instance would have improved efficiency and ease of use. After testing out PASS, about 80 percent of users still considered PASS a very useful tool overall (23 users gave it a rating of above average) with about 90% of these users in a group study setting. Most of the single users gave PASS an average or below average rating supporting that PASS is better used as a group study tool.

Figure 5.1: A histogram showing the evaluation response of 30 students to PASS, 5 of them being single users

**5.2 *Comparison to Google Drawing***

To reiterate, the goal of this project is to replicate the simultaneous sketching aspect of Google Drawing thus it is important to carry out a comparison test to conclude that PASS is successful. The result of the comparison is shown below in figure 5.2.

**Figure 5.2:** A histogram showing the comparison of 30 students of PASS to Google drive based on smoothness, responsiveness, efficiency and overall usability

In this section, a rating of 3 means that PASS and Google drawing are the same in terms of the specified feature. From figure 5.2, it is seen that majority of users cluster at a rating of 3 showing similarity between PASS and Google Drawing. The two features, smoothness and responsiveness in PASS are successful because 100% of users said that it was the same or even better (3 or more). Efficiency however is not a very successful feature of PASS as below 80% of users gave it a rating of 3 or more. Some individuals found PASS easier to use because of its simplicity in connecting users and changing colors. However, efficiency and ease of use is relative depending on the preference of the user. Overall, about 22 users concluded that PASS and Google drawing are the same in simultaneous sketching although some users found it hard to vote overall because of the difference in appearance and complexity.

When asked what they liked best about PASS, the answers converged at the ability for multiple users to draw simultaneously without disruptions as well as the speed in updating sketches on all users’ computers. On the other hand, it was also mentioned that improvements need to be made in the appearance and tools available to users. It is important to note however that the maximum number of students in the study groups for this evaluation was 4.

**5.3 *Responsiveness and Group Size***

To evaluate this section, we started with a group size of two and added new users and observed the responsiveness visually. There was no change in responsiveness for up to 6 users however on the 7th user, we began to observe lagging in time it took to update on all users. By the 10th user, the lagging made PASS almost non-functional. Although, most study groups are smaller than 10 users, accommodating more users will be an improvement to PASS.

**CONCLUSION**

Overall, the goal of PASS was to recreate the simultaneous sketching feature that Google Drawing has implemented. Another main goal of PASS was to make the sketching experience as efficient and convenient as possible. Although results of the evaluation from our test users suggests that PASS might not be as convenient or easy to use as Google Drawing is, many of our test users responded that PASS is in some cases more, responsive and smooth to use than Google drawing. Some of our test users even reported that they preferred the simplicity of PASS, although the opinions varied and nothing conclusive could be drawn from the results. With the indicated goals, PASS is a great success. We were able to make and design a real time, simultaneous sketching program that is relatively on par with the web application created by Google and users confirm its responsiveness. In addition, the response of the usability of PASS from study groups shows that PASS is a huge success with the targeted audience.

The current design of PASS allows many users to draw concurrently, which allows a reasonably-sized group of users to express and communicate their ideas as quickly as the ideas occur to them. Based on the feedback from some of the test users, one future goal for PASS would be to implement additional tools to make certain drawing tasks easier. Google Drawing has several options and toolbars to achieve this, and adding these features to PASS would allow our users to express their ideas even faster. For example, an undo/redo feature, adding an option to insert a table, or adding pre-made shapes (squares, circles, triangles). Another future goal for PASS would be to add options to export drawings into various formats (PDF, SVG, PNG, JPG, etc.) for users that wished to share their drawings outside of PASS software. Lastly, a future goal of PASS would be to accommodate a larger number of users. Sharing ideas should not be limited to PASS software, but PASS is proven to be a good resource for collaborative participation.

**Appendix 1**

1. How would you rate the smoothness of PASS?

Bad 1 2 3 4 5 Excellent

1. How would you rate the Responsiveness of PASS?

Bad 1 2 3 4 5 Excellent

1. How would you rate the efficiency/ease of use?

Bad 1 2 3 4 5 Excellent

1. Do you find that PASS is useful overall?

YES NO

Why?

1. Based on the features highlighted in question 1-4 above, how similar is PASS to Google Drawing

Smoothness

Not similar 1 2 3 4 5 Better

Responsiveness

Not similar 1 2 3 4 5 Better

Efficiency/Ease of use

Not similar 1 2 3 4 5 Better

Overall Usability

Not similar 1 2 3 4 5 Better

1. What do you like most about PASS?
2. What could be improved?

**THANKS FOR YOUR HELP!** ☺

**CONTRIBUTIONS OF TEAM MEMBERS**

**Jairus Bali** did the Processing and Kryonet portions of the PASS software as well as most of the results section of the paper and figures 3.1 and 2.1

**Dara Elebute** did the testing and bug fixes of the software, editing of the PASS Report and background and Evaluation study portions

**Jess Johnson** did the JDBCSQLite portion of PASS software and some Final Report writing sections including but not limited to the software interface and conclusion

**REFERENCES**

**[1]** <https://chrome.google.com/webstore/detail/google-drawings/mkaakpdehdafacodkgkpghoibnmamcme>

**[2]** <http://www.google.com/drive/apps.html>

**[3]** <https://processing.org/overview/>

**[4]** [https://github.com/xerial/sqlite-jdbc#how-does-sqlitejdbc-work](https://github.com/xerial/sqlite-jdbc%2523how-does-sqlitejdbc-work)

**[5]** [https://github.com/EsotericSoftware/kryonet#overview](https://github.com/EsotericSoftware/kryonet%2523overview)

**[6]** <https://javagraphics.java.net>