Server:

The server will hold an arrayList of Strokes. Each Stroke represents the information needed to make one call to drawLineSegment: a start x and y, an end x and y, a color int, and a width.

The server will listen to each user on a socket, and receive a string consisting of each of these ints separated by commas, and then split it into tokens, parse those tokens into ints, and create a Stroke to add to the arrayList. The server will also have a listener attached to this arrayList, and whenever something is added to it, it will send a similar string to each client.

Client:

When the client connects, the server will send current main array line by line. Once it finishes doing so, the client will display the canvas via the GUI. The client will have a send method which will send a string consisting of the relevant integer values separated by commas to the server. The client will also be listening for similar string from the server, and when it receives one, it will call the drawLineSegment method in the GUI.

GUI:

The GUI will listen for mouseDrags, and when it hears one, it will call the send method of the client with the appropriate information. The GUI will also have a drawLineSegment method, which takes coordinates, and ints for color and thickness, and draws a line between the two points.

Concurrency Strategy: Because the whiteboard doesn't need a very high level of accuracy, we want to ensure that during concurrent editing, each stroke is recorded but we do not care about what order they are added. Since we will hold the data for the whiteboard in an array, appending to the array from two sources should not cause issues.

Testing Strategy: We will unit test whatever we can, focusing on the client-server responses. Additionally, we will manually test the whiteboard to ensure that it functions as desired under actual use, with a focus on edge cases.