Project 1

Suppose an application for an opinion poll is to be built. Assume that only one issue is being polled. Respondents may choose yes, no, or don’t care. A server process accepts the votes, keeps the tally in global variables (**yesCount**, **noCount**, **dontCareCount**), and provides the current counts to those who are interested. A client process runs on the client side. The polling is to take place on a university campus during a time period of 30 minutes, allowing thousands of students to cast their votes from hundreds of workstations.

Following is a sequence diagram proposed for the application:



1. Do you agree with this design? **Justify your answer**. If you disagree with the design, provide an alternate sequence diagram that you consider to be an improvement.

The diagram is correct for any individual client; however, it is possible for multiple clients to use the server at the same time. The diagram only accounts for a single client.

1. Is there any need to provide mutual exclusion to the server code? **Explain**. If your answer is yes**, explain** how you would provide that in your code.

In the connection-oriented design mutual exclusion is necessary. In the code it is implemented through java threads. This will allow multiple clients to be processed at the same time.

3. Use Java Socket to implement the server/client software. (Datagram socket or stream mode socket. But have to support concurrency. Complete code submitted in Blackboard.)