The final version of the project has things implemented from the previous submission like:

1. Implemented JOIN and CREATE.
2. Unsuccessful CREATE returns appropriate messages. If successful creates separate thread in which a quiz takes place.
3. Unsuccessful JOIN returns appropriate messages but wasn’t tested.
4. Before starting quiz, inside “group” thread I decided to create temporary file to store quiz received from creator.
5. Quiz is looped so that program consequently reads Question-Answer pairs from the temporary file replacing previous pair.
6. Successfully identifies single winner.

In addition to these also finished on implementing:

1. End of quiz with Showing the results at the end
2. The results are not in descending order (Due to the fact that I was too consumed by debugging all other features and also assumed that it will not be included as in the task of H/W 4, but I guess I was wrong, so that I knew about this objective at very last moment.)
3. Seems to be working correctly with files of various sizes
4. Made CANCEL option for admin, to destroy group before start.
5. All LEAVE commands seem to work properly, while waiting for a quiz to start, as well as during quiz.
6. Timeout works properly (Timeout time might slightly deviate from the hardcoded value = 20) (Sends ENDGROUP message to inform client that he is no longer involved in quiz) (Recorded fd\_set before timer went and then found those who has to timeout by taking all the ones who answered, to the separate fd\_set and CLR-ing form the working one. Then all fds which were in intial fd\_set and are in working one are kicked out)
7. GETOPENGROUPS can be now invoked at any point except during quiz.
8. Admin can now LEAVE during the quiz and it will not affect the quiz which is taking place

The Java Client was modified multiple times to make tests:

1. First modification, CANCEL message now is sent to server when corresponding button pressed.
2. Added readTokenDelimResult(), which is very similar to original readTokenDelim(), to process incoming message of RESULT since original command did not satisfy.
3. Set multiple buttons to be visible in more conditions: e.g. Leave button is now available in Quiz and before it.

To conclude, I followed the path which was developed after at submission 1.5, in which I initially use multiplexing to sort incoming messages of CREATE and JOIN in main thread. When someone tries to CREATE a group, a separate thread is created which will work until the quiz of this group ends. Multiplexing is used again to process commands inside group thread. I also stayed with initial data structures of array of groups[], in which group structure has appropriate info about group and valid bit.