Instructor Dashboard for Monitoring COMPS Collaborative Dialogues

John Carden, MS
Advisor: Dr. Jung Hee Kim
Department of Computer Science
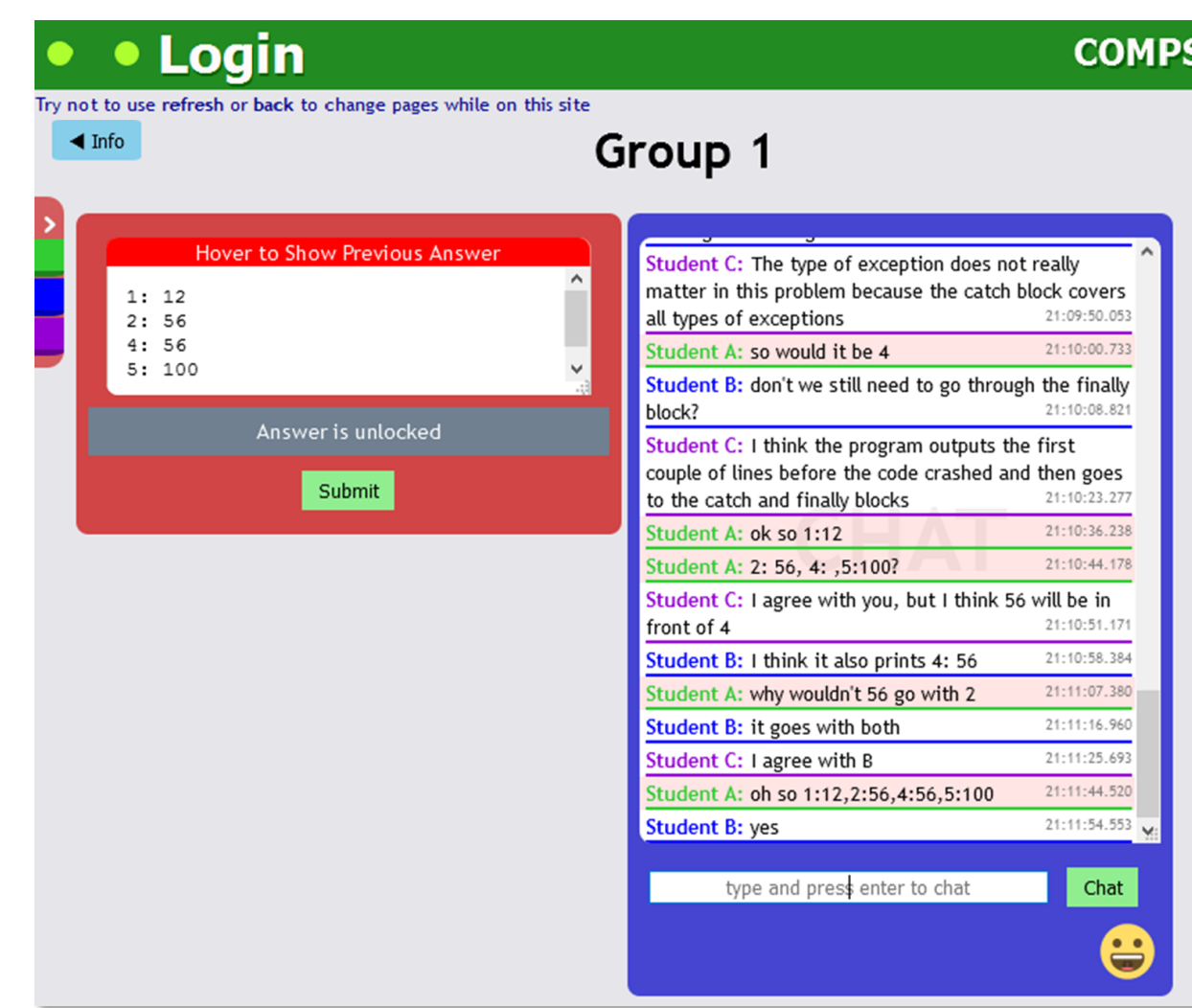
COMPS

COmputer **M**ediated **P**roblem **S**olving

COMPS is a web-delivered computer-mediated problem solving environment to support collaborative learning. It was developed here at A&T.

Students at A&T use **COMPS** for small group problem solving in undergraduate level Java lab sessions. In these sessions, students collaborate together to solve a series of problems.

This results in students learning by doing, and students learning from each other throughout the process.

The interactions in **COMPS** have been used in studies that examine text analytics and the dynamics of collaborative learning.

Project

Develop a dashboard that allows the instructors to see progress and performance metrics for the group conversations, and the individual students. This dashboard should update in as close to real time as possible, and will allow the instructor to gauge the session and solve issues.

The log files are processed every 2 minutes by an analysis engine from Valparaiso U. and NC A&T, which sends metrics back to the dashboard server. Two statistics are currently live:

➢ Text classifiers identify turns where the student seems to have said something containing both content from the problem and other non-problem words. These are evidence that the student is talking about the problem.

➢ Counts of conversation turns are passed through a logistic function so that if every person is participating equally it receives a score of 0.5.

Producing the statistics requires extensive processing. For example certain chat-language phenomena such as using "^" to point to the above turn and *stars* for emphasis are recognized.

Methods

1: Build a servlet to handle communication between COMPS, analysis engine and dashboard client

Analysis Engine

COMPS Server

COMPS Dashboard Client
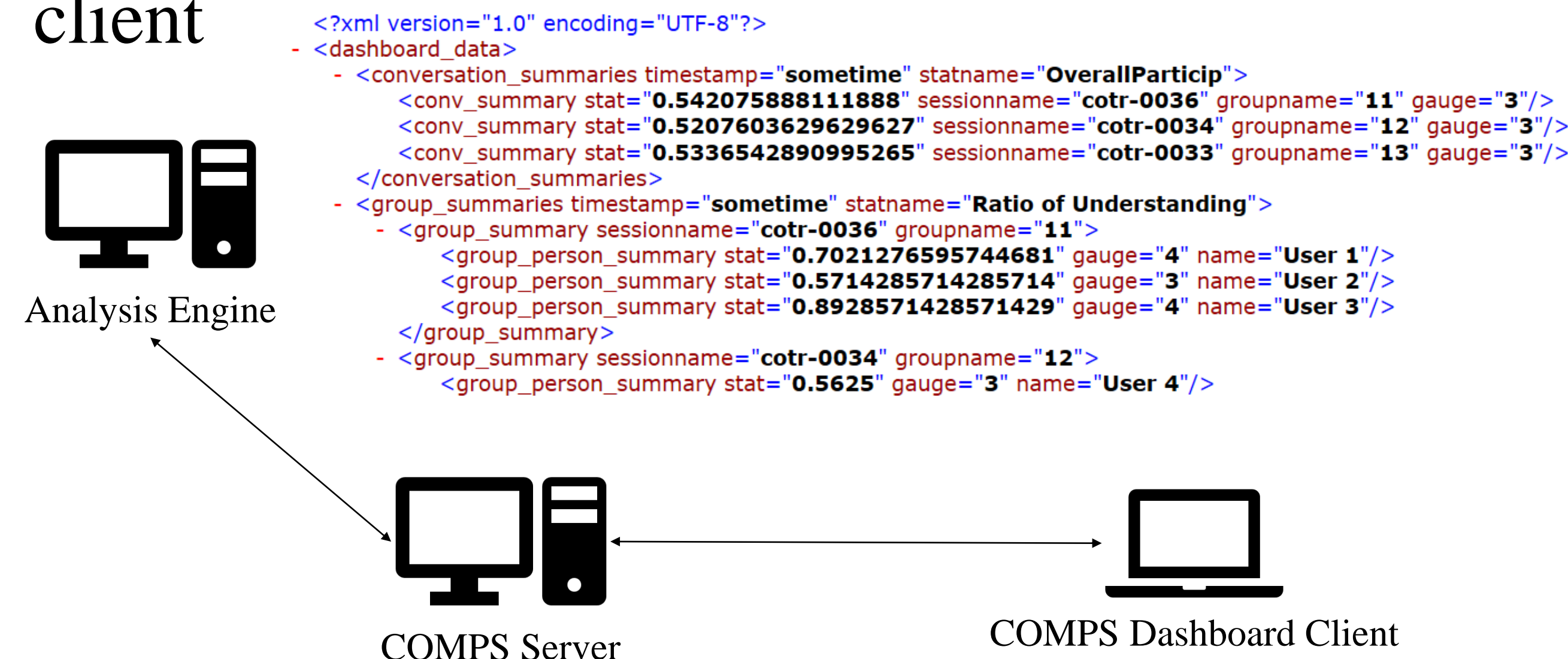
2: Develop front-end design to display information to instructor
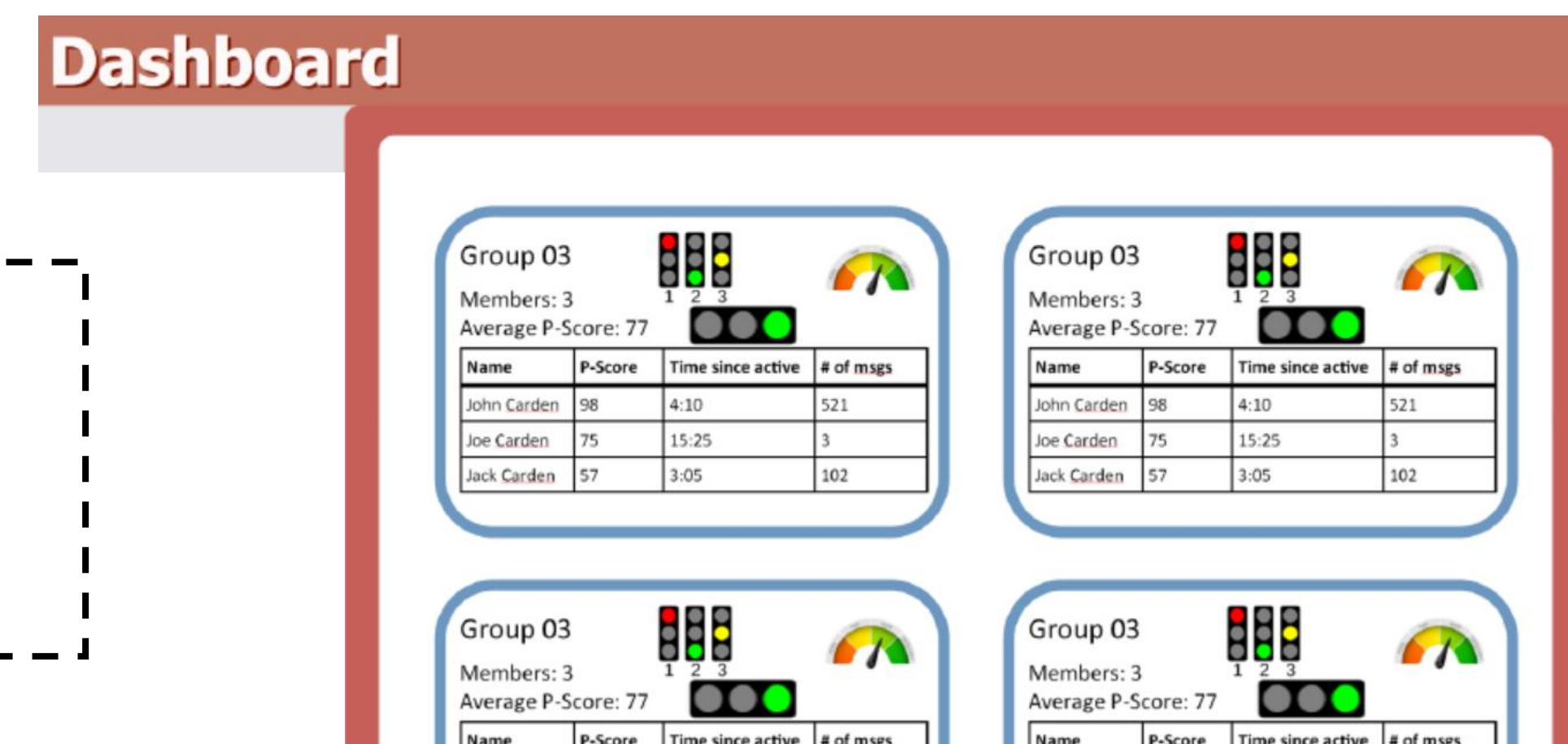
Original design mockup

3: Design logic to process data from the analysis engine and manage the dashboard client

JavaScript enables fast, lightweight handling for client

```
function renderDash(){
  //dashLog("entered renderDash()");
  var oldChatWindows = $(".groupWindow").remove(); //get rid of all chat
  //dashLog(numGroups);
  for (var groupKey in groups) {

    var groupId = groups[groupKey].name;
    //dashLog(groups[i].avgStat());

    // clean spaces
    groupId = groupId.replace(" ", "_");

    // set id to reference table
    var tableId = "attrTable" + groupId;


    var dashWindow = document.getElementById('GroupWindow'+groupId);

    //dashLog("Checked for window");
    if (dashWindow==null) {
```
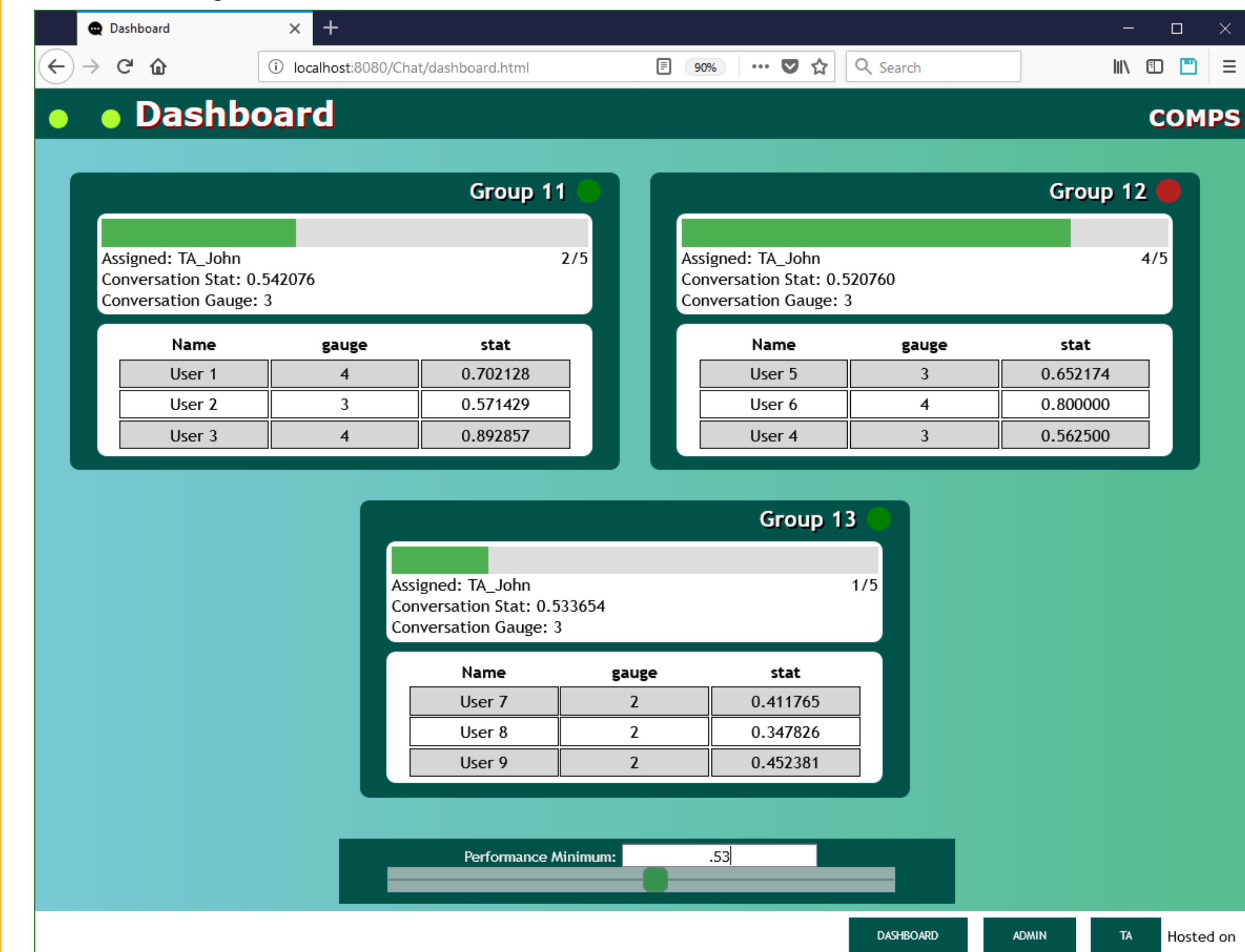
Results

Trial run in a real class setting in Spring 2018:
➢ 23 conversation groups, 75 minutes each.
➢ About 70 students.
➢ Conversation status based on standard deviation of the different students' participation (a higher std. dev means somebody is dominating the conversation).
➢ Student individual status based on how often they stated meaningful dialogue turns.

For ease of use, the final dashboard (pictured here) has been designed to dynamically resize itself if more metrics are added in the future.

Future Work

Next steps for this project will include additional metrics and testing to improve the accuracy of the metrics displayed. The current metrics are a work-in-progress. A new metric will be studied that tries to measure how frequently students address each other, which is likely a sign that the conversation is or isn't being productive.

Acknowledgments

7th Annual COE Graduate Poster Presentation Competition