Engagement-

Studying the various problems of Attendance Management, a repeating factor in grievances was lost Classroom time. The obvious solution is to minimize as much time as possible. However, many suggestions from professors in our survey revealed an opposite approach. We can let Attendance take up class time if it’s a part of class and can help with student engagement in the classroom. The key focus points of this application are to track attendance first, and measure or encourage engagement second.

To track attendance through the lens of engagement, the ways engagement could be tracked were considered in designing this application. Many professors use attendance as a participation grade. Quizzes are occasionally given out at the beginning of classes to serve either as a reminder of homework or a previous class or a test to see if students payed attention or completed homework. These could vary between multiple choice easy to grade quizzes, and open ended questions to spark discussion and interesting ideas. One professor in our survey specifically asked for a way to easily cold call students for discussion.

Considering these points, as well as those from our previous reports, the initial non-functional requirements of this application needed to be that it was portable enough that students could use it on their laptops, and must be fast and easy to use for both students and professors. Functional requirements for the final product are as follows. The interface will separate which users are professors and students. Professor abilities will include setting up quizzes and their duration, type, questions, answers, and deletion. They will also be able to see past quizzes and their results. Student attendance can be viewed by individual student or by class as well. Attendance determined by whether quiz was taken. Student key requirements however only include the ability to take quizzes open by class. An additional functionality for the student would be to help the student keep track of miss days or quizzes in this case. Because the professor would set the duration, it would encourage the student to be on time.

The application currently written in Java using Netbeans IDE, contains an object oriented inspired design for fluidity in the interface. Its backend Database is MySQL, and has ten tables holding data. Students and Professors upon loading the program need to login and will see similar interfaces. Both have a section that quickly states whether there is/was a quiz today, and options for the professor to see results of the quiz, or for the student, if its open, to take it and submit. Professors can search by date for past or future quizzes and set up new ones. Past quizzes are not editable, future ones however are. Currently, only default quizzes are set for creation. Default quizzes contain one “I’m here” button that creates the log that that student took the quiz and was in class. For a particular quiz, a professor can quickly see all who participated, and click one button to randomly call a student. A future functionality once non-Default quizzes are enabled, is to let the professor see a spreadsheet of answers, without student names, for open discussion.

Two major drawbacks to this implementation are the java interface used, and its reliance on the honor system in students. The Java GUI used while portable is not as preferable as for a web scripting language like php. Quizzes could be taken in one’s room, and students could then claim to be in class. However, the engagement and its use in class discussions would deter those from cheating the system.