**Project Deliverable #5: Second Release Report**

1. The customer asked us to implement the **multiplayer** and **chat** user stories. We implemented these stories. We spent about 40 hours total working on the project; compared to our estimate, we within an hour discrepancy.
2. The Extreme Programming (XP) principles/practices that we followed were the same as with the previous release: incremental planning, small releases, refactoring, pair programming, collective ownership, and sustainable pace.
   1. Incremental Planning – this was practiced in the form of the second deliverable our group submitted. We followed the guidelines to list our stories and describe them, including the user tasks and non-functional requirements. Rather than recording the requirements on story cards, we recorded them in a single word document (project deliverable #2).
   2. Small Releases – this is required of all groups in the class. We are all required to submit new releases/iterations of our projects about every two weeks. Our group decided to develop the single-player function for the trivia game, which is the bare minimum needed to play the game.
   3. Refactoring – after we coded things like the layout of the pages, we cleaned things up a bit. The images for the project were originally all in the root folder. We created an image folder for them, and then had to alter the code a bit to find the new location of the images.
   4. Pair Programming – we split into pairs, with one person of the pair coding, the other checking. Then the other would code and the other would error check. One pair worked on the php, the other on the JavaScript. Then the pairs swapped assignments. We went from there.
   5. Collective Ownership – we made sure that no islands of expertise developed by doing the steps described in part d.
   6. Sustainable Pace – our group did not work past 10 PM each day of work. And we did not exceed 8 hours of work per each workday.
3. The test case for account creation was to make sure the account could be created successfully and stored in our database. The inputs are “username” and “password”. The expected output is a message across the top of the page that says “Created account successfully” if the account was created, “Please fill in all fields to create an account” if the player did not fill in all fields, or “Username is already taken” if the username they entered has already been created.

For the single-player story, we tested that the trivia game could be played; that is, a random question is presented to the player with four choices to pick as the correct answer. After submitting their answer, an alert pops up and tells them whether their answer is correct or not. Then a new question is presented and the cycle continues. The input is the players answer choice. The expected output is an alert box saying “GOOD! Correct answer” if they answer they chose was correct, or “TOO BAD! That’s not correct!” if they pick the wrong answer.

The user-submitted questions were tested for whether or not they could be stored correctly in the database and then generated for the player to answer during single-player mode. The inputs are “Question”, “Correct Answer”, and three “Fake Answer” fields. The expected output is “Congratulations! Your question was successfully submitted!” displayed at the top of the page if all fields of the form are filled. “Please fill in all fields to submit a question” is displayed if one of the fields is empty.