



## TEAMWORK REFLECTION

### Evaluation

*Evaluation of project from team planning perspective*

We are all pretty happy with the way this project turned out. Initially we were going to attempt to get some extra features done such as text confirmation and some statistics visualization but we ended up narrowing our focus and making sure that the code was as robust and reliable as possible.

Through the project we also came up with some nice ideas that we were able to implement, such as pictures for each group, and things that we want to do in the future, visualizing player statistics and more sophisticated ranking algorithms.

We had a reasonable scope from the beginning and we were able to finish the core of our plan and we did it well. Our application will go live this week and we are throwing a launch party on the night of demo day.

### Lessons learned

*Summary of key lessons learned*

- Do not underestimate the complications small features can add
  - After creating the MVP we had to add features to flush out the user experience. Adding these features gave us the complication of keeping our site clean while making it more functional. The groups feature in particular lead to a massive change in our code that was slightly unexpected.
- Use github features to make your life easier
  - Rather than ignoring all the features github provides, use them!
  - From pull requests to issues, github provided a way for us to communicate what we need to get done and how we were going about doing it.
  - This prevented a lot of confusion and code breakages.
- The importance of planning
  - There were several features where we decided to try and implement it before

planning it out and all those attempts did not go well.

- In the future, we will spend more time thinking about how a feature will work and plan out the implementation before diving into the code.
- Go with the simplest solution
  - This may seem like it contradicts the importance of planning but the two complement each other. There were several times where we planned for a feature to be more complicated than it needed to be in attempt to scale or while thinking about the future. This led us to create bulky features that were hard to iterate on and we ended up deleting a lot of code.
  - In the future we will try to choose simple solutions that get the job done. We can then iterate on these features when we need to scale.
- Communication!
  - This is key to a smooth coding process because it keeps you focused on what you need to do while preventing you from accomplishing redundant tasks.
  - We used github and common social networks like gmail and facebook to keep in constant contact.
  - Keeping in contact also prevented people from falling behind in the code and helped motivate each of us to work together (both at the same time and in general)