

## Release 1 Team Reflection (Summary)

Team	g262-01a The Squad
------	--------------------

**To what extent did you abide by the assigned roles, and, in general, how are the roles working out? Did anything fall between the cracks because a role was missing or not done properly? How would you adjust roles to make things go more smoothly?**

For the most part roles were ignored and everyone took initiative where needed. All in all this worked well because everyone had an equal stake in the project as well as an equal stake in the design process ensuring that everyone understood every part of the system. A hierarchy would have impeded the flow of ideas and concerns which have pointed out a few major issues early on in the process. That being said we did in fact allow certain things to slip through the cracks such as the completeness of our documentation and similar submissions. In this respect it would have been useful for a single person to have the responsibility of checking everything off before handing in. This is something we admit we should look into as well as assigning similar small responsibilities for each member. We also see the roles defined in the beginning of this project coming more into effect for revision 2 as we begin to require more from our testing coordinators.

**Was the workload divided appropriately? Did everyone contribute to the product itself (e.g., did everyone participate in programming and construction)? How would you adjust the assignments and responsibilities for the remainder of the project?**

The workload was divided pretty evenly across the group. Large sections of code were done by people taking initiative and doing what they were already good at. This self-assignment allowed the best person to be put on the best assignment that fit them. This process was loosely regulated by the development coordinator that helped make sure we were on the right direction but the majority of the work was self directed. If anyone got stuck on something because a feature was not implemented yet someone who was less busy stepped in to add the missing feature and make sure development could continue. With time everyone developed their own specialization such as Forms creation, database handling, front end development. This is a necessary step in making sure the team has the necessary skills to finish the project. Instead of each person having to overcome the same problems once one figures it out he communicates it to the rest and makes sure that he's there to help if a problem they already encountered comes up again.

**Were there any unintended consequences of your plan, your design, or your work? Were any pitfalls encountered; did any of your identified risks come to pass? How might you plan for and deal with the unexpected in the remainder of the project?**

The biggest unintended consequence was planning too detailed too early. Before most of us had a good grasp of how django handled databases we planned our entire system with a traditional sqlite backend and intended to interact with it using straight sql queries. Its now apparant to us that we should have made sure we all finished all of the tutorials before designing the system. Starting the process before understanding the development environment we were working with was a mistake and ended up costing us a lot of time.

The reason for entering such detailed planning so early was an attempt to mitigate the risk of not planning carefully enough and having to restructure our entire database and consequently the system itself mid production. Unfortunately by trying to avoid this exact situation we ended up driving right into it. Luckily we managed to cope with this by diligently keeping track and changing our class diagram with every alteration we made to the architecture. that way we didn't stray too far off what the design document described and if you ever were unsure of how things connected our diagram always matched the current status of the database.

**Are team meetings effective? Why or why not? Does the team believe it needs to restructure the way its meetings are run? In particular, do you need help in making your meetings go better?**

Team meetings were extremely effective in organizing the team. When we tried to do things remotely not everyone was as focused and it ended up adding more confusion than anything else. That being said we also noticed that short team meetings were not very useful in actually getting things done. They might have looked that way but in the long run actually hurt us because we would never finish everything we wanted. After we switched to much longer development meetings where all of us are working face to face development went by much faster and allowed us to plow through any road blocks we faced.

**What is the most important change you can make to perform better (as a team) in the remainder of the course? Can you improve your process? Can you improve any of the technical artifacts you've created?**

The most important change we have done right before our first revision was having team development time where all of us get together and program next to eachother. This change has completely changed how much time it takes us to develop parts of the site. It has also allowed us to implement pair programing that helps all of us get over large obstacles. We can definitely improve the process by defining more clearly what we want to get done in each development sprint (each week).