Sprint Retrospective

## What went well?

Scrum meetings went well, they started at the same time every day (9:20 am), unless we had a lecture.

As well as scrum meetings in the morning we had a group summary at the end of the day where we showed off what progress we had made and what we needed to do tomorrow. This gave us a good mind-set and kept us in the loop, so we knew how every area of the project was coming together.

Towards the end of the sprint, we were communicating quite well. The team knew what other people were working on and what there are of expertise were which helped with the structure we were seemingly lacking. This allowed us to ask questions to the right people.

GitHub usage was pretty good throughout, we didn’t run in to any problems merging people’s work. Branches were used to reasonable effect here. After a few days we started code reviewing pull requests. We made good use of the issues feature on GitHub which we would assign to team members and they would be used to monitor the progress of the project as there was list of issues making up a to-do list which could be moved to ‘In progress’ or ‘Completed’ as needed.

The website is not finished but ended up closely resembling the wireframe and our original ideas with only minor tweaks. Our clients during the meeting liked the design of our website which was definitely a plus given the client-oriented nature of the project.

The pair programming almost felt natural as we helped each other out to problem solve our code.

We agreed a lot on how we wanted our web application to turn out in terms of the layout and the way we would make things work.

## What went wrong?

We felt a little bit lost in trying to estimate how long tasks would take us partly because it was necessary to do some research in order to be able to complete certain tasks.

In addition to this, using story points added to the confusion of judging how long tasks would take in large part because it is tough to judge exactly the time per story point and every member will have a different number of story points that they can do in a set amount of time. So while team member A could output 1 story point per hour; team member could output 3 story points per hour, thus making it difficult when deciding how many story points it would take to complete the task – one could perhaps note down 6 story points from team member A or 2 story points from team member B, but in the our roles ended up being more flexible than first thought, as multiple people sometimes worked on the same task or another person ended up working on a task that we thought a different person would have worked on.

It took a few days before we got a clear idea of the project and its many components which meant our progress was somewhat stunted for the first couple of days.

We didn’t really update the sprint backlog for the first two days, so we had to spend some time reconstructing what people worked on from looking at GitHub and Scrum meeting notes. This was compounded by the fact that we worked on the backlogs sporadically and locally (did not push a new version to GitHub daily) which could have been avoided.

## What can improve?

Sitting down and working more as a group on keeping the agile documentation updated. We could plan to do this daily after the scrum meeting.

We should pay more attention to working on specific ‘tasks’ up on the GitHub projects board, update them during the week and reference them in any pull requests.

We should work on our tasks on separate branches, and merge these into a development branch whenever we reach a significant milestone on our task. This would force code review for all features written by the group (Some people were working on the branches we were merging into, so their code ‘escaped’ review). One particular example of this from the feedback video was about scrum meeting notes (readme file) were updated directly on GitHub which pushed to the master branch which is not a good practice so those changes will be from now on pushed to a separate branch first then merged in to the master/dev branch.

The sprint burndown chart needs to be modified because we were unsure of how it worked exactly (with regards to inputting values into the spreadsheet – how many story points remaining) so we will focus on adjusting the values in order to fix the chart which will be beneficial as it will show us the pacing of the first sprint, thereby telling us if we started too slow/fast, sped up/slowed down as the week went on etc. Though as noted in the above section we started off slowly as we had to do some research on certain elements of the project (e.g. Java Server Pages) and setting up the database and importing the database took longer than we first anticipated, so the product features ended up being neglected for the first couple of days.