Actions based off of the Sprint Retrospective - Report

# List of actions:

1. Focus on updating the documentation daily rather than sporadically and upload each new version to GitHub rather than working on it locally as that does not provide evidence of our daily work.
2. Tasks should be worked on in new (for new functionality/milestone), separate branches before being merged into the main branch.
3. Try to force more code review by increasing the number of required reviewers from 1 to 2 which means there is more potential for comments/feedback from other group members. Naturally we aim to take a better look at commits involved in a pull request and always leave a comment, suggests changes etc.
4. When making a pull request we should reference a specific issue/task so we can more easily track the progress of each individual task. If the issue does not currently exist then it should be created first.
5. Fix the previous Sprint Burndown chart and keep the second chart up to date so it accurately displays our Sprint’s progress.

# How these were carried out during the 2nd Sprint

1. We did a much better job in updating our documentation and did so every day. It became a scheduled task after our scrum meeting that we would update the sprint backlog, product backlog (if changes could be made) and scrum meeting notes. We found that we had a greater understanding of whereabouts we were progress-wise in the project.
2. On the whole we made improvements in our use of branches – creating a new branch when working on new functionality and resulting in a pull request when looking to merge into the main branch. So for those times fewer issues would arise as merging was done more safely. However, there was a duration where two people were working off of the same branch and would be pushing directly to it which ended up causing issues when pulling the branch.
3. Code review was used to much greater effect in the second sprint in large part because (relating to action 2) more pull requests were being made as a result of creating new branches that would be merged in to the main branch, so there ended being more opportunities to review code that may otherwise have escaped review (being pushed directly). The quality of the code reviews increased because we changed the number of reviewers from 1 to 2 which meant more feedback was provided and in multiple instances change requests were made because a reviewer found something that should be altered in some way.
4. When making pull requests we would make sure to reference the GitHub issue that the pull request was related to (sometimes it could reference multiple issues) which allowed us to keep better track of the development of individual tasks in the project/sprint. This was not just limited to proper tasks, it could be bugs, documentation etc.
5. As part of a larger effort to bring all the first sprint’s documentation up to date, we fixed the burndown chart to show the total amount of work left to do (day-by-day). Then we made sure to keep updating the second sprint’s burndown chart every day with the values being added properly this time and this was very useful in showing us how much work remained, and if we had to speed up or not (we had a fairly consistent level of work throughout the week), and coupled with GitHub’s project progress bar it proved very helpful.