CS21 Retrospective Evaluation

The purpose of this document is to periodically look back at the problems raised in our retrospectives and ensuring that we are putting the plans we made into practice. The evaluation section is to reflect on our actions and discuss whether we have learned lessons from previous sprints

Retrospective One:

The first retrospective was conducted verbally on Microsoft Teams which was in hindsight not the best way to document our findings. We hope to adjust this process by using an online sticky note app to mimic the way we would hold a retrospective in person. We didn't have too much to talk about since the project is only just beginning, and this sprint has only been used for planning and designing. Despite this, we identified the following issues:

Problem	Our communication has not been very good during the first few weeks. This is understandable given the current circumstances – it is not easy to gel as a team when we have not met each other in person.
Action	Every message on Teams must receive a response from the necessary people. This does not have to be a reply, but could just be a like, to highlight that the message has been received and is being taken on board.
Evaluation	Looking back at this problem, we have been improving our communication. At the weekly stand-ups we regularly discuss whether we are happy with our communication and whether the protocols we devised are being followed.

Problem	We have not been fully utilizing Gitlab by adding issues.
Action	Ensure to add an issue for every job we do in the project going forward.
Evaluation	When we look back at our new Gitlab issues, it is clear that we have been working hard to use them more. Additionally, we have made sure to assign people to issues and tag them with important information such as urgency or area of development.

Retrospective Two:

Problem	Be more specific with issues, rather than overarching tasks.
Action	Make more issues for complex tasks.
Evaluation	On reflection, we have been trying to improve our issue selection. Instead of simply translating our sprint goals into large issues, we have been making a more conscious effort to make more precise tasks. Perhaps this could be done more effectively, and there is room for improvement in the future.

Problem	Continue improving communication on teams.
Action	Make sure that we keep track of teams to keep following the communication rules we set out in the last retrospective.
Evaluation	We are happy that we are making the most of teams to communicate.

Problem	Continue release branching strategy
Action	N/A
Evaluation	When we look back at our repo, we can see each individual release that we made. We are happy that we have made this choice as it has helped to prevent large merge conflicts. The last thing we want to be doing is wasting our time sorting out merge conflicts, so we are more than happy to continue with this strategy.

Problem	Continue work split-up.
Action	N/A
Evaluation	Our issues show that we have been splitting the work up between ourselves. Each issue has been assigned someone to work on it. Perhaps some issues may be tackled more effectively in groups, helping us to learn off each other. This is tricky given the fact that we are all in different places across the globe, but we will hopefully try to implement some group work in the future.

Problem	Stop committing to master when there is a specific release branch.
Action	Put new artefacts and updates into the current release branch instead of master.
Evaluation	The repo is evidence that we have been following this rule. As a result, our lives have been made much simpler when it comes towards the release day and we want to merge our work.

Problem	Stop committing too many times and committing files which should be in .gitignore.
Action	N/A
Evaluation	There have been quite a few commits, specifically by David which are not necessary. It is hard to strike a balance between making commits which are small enough to aid with merging, but also big enough so that we are not committing too many times. This is something that can be improved upon as the sprints continue. We have updated the gitignore to solve the second issue.

In general, the second retrospective was helpful but could have been slightly more effective. For example, three of the issues were regarding areas we want to continue working on. It might have been a better use of our time to delve deeper and identify areas of weakness within the process rather than areas of success. As a result, we will try to be more critical of ourselves in the next retrospective.

Retrospective Three:

Problem	Start giving weekly update to Hayden.
Action	We discussed this issue as we felt that it was a long time between customer days, and we did not want to leave Hayden in the dark for too long. We did, however, come to the agreement that every week was potentially too frequent. Instead, we would make a better effort to communicate with Hayden via email, in between customer days, but with no set time.
Evaluation	We have done well with this issue because Ans, our product owner, has been overseeing the process. Ans sends emails in between customer days, making sure that Hayden is able to attend customer days, if any of his colleagues will be attending, and whether we require a more casual meeting with him in between.

Problem	Make more common merges into release branches.
Action	N/A
Evaluation	We have been making an effort to do this since large features will cause large head aches with merge conflicts. With this comes the possibility of committing and merging too regularly so it is important to strike the balance.

Problem	Start writing more descriptive commit messages.
Action	We agreed to follow the format discussed by Tim in the lectures.
Evaluation	This has been followed well by everyone as can be seen in our repo. Perhaps it would be beneficial to make a template commit message and store it in our wiki for reference. In practice, we have realized that every commit is a snowflake and the template will not always fit. We have, however, stepped up our game in terms of being more descriptive, maybe just not always following a set format.

Problem	Continue presentation format.
Action	N/A
Evaluation	This format has worked well at each customer day, with praise from both the customer and the marker.

Retrospective Four:

Problem	Start experimenting with different programming techniques like pair and mob programming.
Action	We agreed that we would start off trying pair programming with Ans and Lewis collaborating on the feature of operator API post requests.
Evaluation	This has been a worthwhile experience. Both Lewis and Ans have been able to learn off of each other. It was a good idea pairing Ans, who has done work on the API, with Lewis who has focused more on the project management and version control. As a result, they were able to learn off each other and solve problems quicker than usual.

Problem	Start thinking about risks associated with all our project features.
Action	N/A
Evaluation	This has been something we have been trying to discuss at each meeting we have. At the end of each meeting, we leave the floor open to any issues we think might arise or any problems we have. Additionally, we met with CS20 to discuss and mitigate the risks associated with potentially combining our solutions for Hayden.

Problem	Start being less rigid post customer days, not getting bogged down in our issues.
Action	Try and hit the ground running so not to waste time.
Evaluation	Reflecting on the beginning of the sprint, I think we got lots of work done. We had to since we gave ourselves a lot of work to pull off. As the February customer day is approaching, we do not have as much urgent work to take care of as we have done on previous customer days.

Problem	Start putting more emphasis on testing
Action	We assigned Pragati to be our “test manager” ad oversee all testing on the project. Each of us will individually have tests to write, but Pragati is in charge of coordinating and running these tests.
Evaluation	It has been good to have a team member overseeing the issue. As a result, we have been creating lots of tests and hope to run them automatically soon. Unfortunately, we are still struggling setting up the CI/CD pipeline.

Problem	Continue branching method – i.e small manageable features.
Action	N/A
Evaluation	Looking at our repo, there have been more branches made and merged throughout the sprint. I, David, have been making sure to make smaller branches and merge them into release-four regularly to avoid merge conflicts later on.

Problem	Continue meaningful commit messages.
Action	N/A
Evaluation	As with the last sprint, this has been going well.

Problem	Try to make less changes in the code in order to reduce merge conflicts.
Action	N/A
Evaluation	This has been something which is tricky to do in practice as we are always trying to improve upon our code, especially after the code review we conducted at the end of the last sprint. However, we have been trying to make these changes in small branches, as detailed above.

Problem	Continue improving communications on teams.
Action	N/A
Evaluation	Despite the fact we have already discussed this, we felt it was important to maintain the standard we had set. Communications had slipped very slightly in the last sprint, which is understandable due to the holidays, but we hope to improve upon this in the future. It is important not to forget about the issue we raised early on in the project.

Problem	Continue logging work completed in order to help write dissertation.
Action	Keep a progress log of everything we have done each month and any issues which came up.
Evaluation	This log has been kept on teams so that everyone can have an input. It has been a good idea as it will help us to reflect more effectively in our dissertation.

Retrospective Four:

Problem	Pythonanywhere isn't very cooperative with the tools we are using.
Action	Yes, Pythonanywhere is not compatible with the CI/CD pipeline and we can't clone our repo to our account, but these are just minor issues. The only reason we hosted on pythonanywhere was to demonstrate our proof of concept to Hayden which has been done. We do not anticipate a great amount of work being required on it in the next sprint so we have agreed that it would be more bother to change our hosting platform at this stage.
Evaluation	This was a good decision. Python anywhere hasn't really been required so far in this sprint so hasn't been causing us any problems.

Problem	Secure communication with CS20 web app
Action	There was no predefined security protocol for the API spec to guide us, so both teams made different protocols. This means they were not compatible. We decided it may be worthwhile to disable our security protocols and just prove the communication with the other team via the API.
Evaluation	We are currently in talks with CS20 to arrange a time to meet and discuss this.

Problem	PAS212 and Post/Put requests
Action	No action needed as the problem was solved during the last sprint.
Evaluation	N/A

Problem	CI/CD Pipeline not up and running
Action	Keep Lewis as the lead on this area, with others providing assistance if required
Evaluation	We have kept Lewis focusing on this area. The rest of the team are not very experienced with CI/CD pipelines, so Lewis would have to explain the process to us, which would be a waste of his time. Instead it would be more productive for one team member to work on it.