Retrospective A

COMP3900 2022 Term 2

Computer Science Project

Project Title: Dinner Party

Group name: NewWorld

Retrospective Meeting Date: 1st July 2022 12:00PM (Sprint 1 submission date)

Submission date: 10th July 2022

Attendees:

Antoinette Ayoub (z5254617) Scrum Master, Developer (present)
z5254617@unsw.edu.au

Tom Killingback (z5256086) Developer (present)
z5256086@unsw.edu.au

Tiger Liu (z5160350) Developer (present)
z5160350@unsw.edu.au

Zico Chen (z5248252) Developer (present)
z5248252@unsw.edu.au

Derek Chen (z5289988) Developer (present)
z5289988@unsw.edu.au

What Went Well

Item	Description
Visual Design & Wireframes	The Figma wireframes that were created for the proposal were of a high visual standard and served as a great reference for the frontend team to work towards. This early inception of a well presented product goal was good motivation for the team and improved the project's aesthetic quality. As the flow of the application was already agreed upon, there was consistency in the design and usability of the frontend.
MongoDB	This choice of database was easy to use, well documented and allowed easy integration with Python and Django (forming the backend stack). The web interface of MongoDB Atlas made it easy to visualise and edit data.
Git Practices	Use of rebasing was a good practice, which ensured all members maintained a current understanding of the code base. Despite some rebasing issues towards the start, all members learned how to rebase properly and this improved over time. Additionally, an enforced rule for independent reviews of PRs was a great opportunity for team members to share knowledge, provide ongoing feedback and gain exposure to work outside of their primary domain.
Product achievements	All sprint issues/stories were successfully achieved to the expected quality. Despite a steep learning curve, completion of the relevant issues which formed the first sprint provided the project with a strong foundation and good start.

What Did Not Go So Well

Item	Description
Learning curve (Django REST)	When selecting frameworks in the tech stack, a key consideration was the learning opportunity which they brought on. However, this posed a steep learning challenge, which added more complexity than predicted. Given the short sprint time frame, this added increased workload and stress. The Django REST framework was new to all members. The backend developers were too keen to begin work and skipped a lot of foundational knowledge. Hence, the learning curve became apparent towards the end of the sprint during critical phases, causing delays to their dependencies on the frontend.
Learning curve (Styled Components)	Similarly, React Styled Components was new and posed a steep learning curve for some frontend developers. As the Figma wireframes were very specific, styling became problematic and was very time consuming. Ultimately, the challenges associated with the use of styled components were overcome, but this was at the cost of significant time which would have been better spent elsewhere.

Lack of communication	Communication breakdowns caused dependency issues, where a certain feature would fall behind schedule and impede the progress of features, which relied upon it (e.g frontend waiting for backend endpoints). Members were not actively communicating these delays, so these issues were not identified until they became problematic. Furthermore, some members did not properly communicate their unavailability. Thus, when issues were encountered, they were not able to be reached. In certain cases, they were the domain expert and it became extremely difficult to continue work in their absence.
Poor scheduling	Although daily synchronous standups were scheduled, poor availability proved to be a barrier for this goal. This resulted in desynchronisation, which contributed to the poor communication as previously discussed. A result of this was coding efforts happening in short bursts and sometimes overnight, instead of being distributed throughout the allotted timeframe. This was unideal as it meant a lot of work was completed with little time to spare, leaving little room for error.

Things to Try in the Next Sprint

Item	**Description**	**Action Owner**
Incremental Deadlines	Smaller and more incremental day-to-day deadlines will be imposed on each task, to ensure progress is made more consistently. Furthermore, failure to meet these deadlines will highlight issues earlier on and allow them to be communicated and addressed accordingly, instead of being detected at the last minute.	Tom
Communicating availability in advance	In order to cater to all team member's varying commitments and workloads, regular 'when2meets' will be sent out to plan working sessions and sprint meetings instead of sticking to a fixed time slot. This will allow for better utilisation of availability and increase attendance.	Tiger
Standup structure	Standups will have a fixed and consistent structure, so they can be easily read and interpreted, allowing for a better understanding of each other's progress. Standup structure: ● What I've done today ● What I plan to do tomorrow ● Any barriers or concerns	Antoinette
Centralised communication channel	All communications should be centralised in one location, which will be Slack, as opposed to a combination of platforms as previously used. Ensure all members have notifications turned on and that Teams chats are not used for communications to ensure that all information can be found in one location, instead of being scattered around.	Antoinette

Pair Programming	Where possible, organise for pair programming to address issues/stories to reduce extra time allocated conducting asynchronous code reviews. Furthermore, this will ensure all members are more up to date with the codebase and will increase the quality of code by using a second opinion.	Derek
Task assignment and actionables	During meetings, all tasks should be explicitly assigned to assignees using Jira. All tasks should have an assignee and should be displayed on Jira. If not, an additional “bug” or other relevant issue should be created to track the actionable. This will increase accountability for actionables, as previously, issues would be raised, but would not be acted upon as nobody was accountable and these were not explicitly tracked.	Zico

The action owner is responsible for attempting to enforce and follow up on their assigned items.