

# Retrospective B

## COMP3900 2022 Term 2

### Computer Science Project

**Project Title:** Dinner Party

**Group name:** NewWorld

**Retrospective Meeting Date:** 22nd July 2022 (Demo B date)

**Retrospective Meeting Time:** 11:00AM - 12:00PM

**Submission date:** 28th July 2022

## Attendees:

**Antoinette Ayoub (z5254617)** Scrum Master, Developer (present)

[z5254617@unsw.edu.au](mailto:z5254617@unsw.edu.au)

**Tom Killingback (z5256086)** Developer (present)

[z5256086@unsw.edu.au](mailto:z5256086@unsw.edu.au)

**Tiger Liu (z5160350)** Developer (present)

[z5160350@unsw.edu.au](mailto:z5160350@unsw.edu.au)

**Zico Chen (z5248252)** Developer (present)

[z5248252@unsw.edu.au](mailto:z5248252@unsw.edu.au)

**Derek Chen (z5289988)** Developer (present)

[z5289988@unsw.edu.au](mailto:z5289988@unsw.edu.au)

## What Went Well

Item	Description
Centralised communication channel	Slack was better utilised during this sprint, as the team agreed on allocating a thread to each issue. This meant that looking for historic communications and decisions from other team members was made easier, reducing any confusion from conflicting or scattered communications. Additionally, all team members had access to all communications, which resolved the issue of missed communications when previously using personal messaging services.
Endpoint planning	Before commencing work, the team had a synchronous meeting and planned the schema for endpoints in Confluence, so that everyone would be on the same page and have a chance to voice their inputs. This was especially important as members were working across the stack. By doing this planning first, less synchronous communications were required, as everyone had a clear and unified reference to use, which boosted efficiency.
Standup structure	Tom implemented the Alice bot on slack, which involves automated standup prompts with the questions “what did you do today”, “what will you do tomorrow” and a third “questions/queries” section. These responses were shared to the #standup channel. These encouraged and structured standups helped increase visibility across the team, as team members would have to report daily. This assisted in keeping one another accountable and on track. This method will continue to be used into the future.
Pair programming	By promoting pair programming, members were able to synchronously review each other’s work and keep track of progress. This way, backend and frontend developers gained exposure across the stack and better understood the complexities of one another’s work. This was also an effective way of reducing bugs by holding one another to a certain standard and also allowed developers to learn from each other too.
Task assignment and actionables	By breaking issues down into atomic sub-issues, these tasks were able to be assigned as issues in sprints. Additionally, all actionable items were strictly converted into issues to maintain visibility and accountability. Utilising the Jira board, this made tracking actionables a lot easier. Furthermore, as each actionable had an assignee, it was clear who was responsible for completion, meaning reaching the correct person was easier too. This also encouraged the assignee to ensure their task was completed by the due date.
Communicating availability in advance	Tiger was very proactive in sending out when2meets at the beginning of each week and sprint to coordinate sprint meetings around the team’s diverse and busy schedules. Consequently, meeting attendance was improved, which resulted in the team being a lot more synchronised, reducing the need for blocking asynchronous communications. Furthermore, this assisted with supporting the pair programming initiative.

## What Did Not Go So Well

Item	Description
Incremental deadlines	The team attempted to promote incremental deadlines and whilst this assisted in identifying issues earlier before they became blocking, there was still insufficient availability to rectify these identified issues. Effectively, team members with limited time and poor availability were unable to complete their tasks during their allocated times, which blocked progress of other issues. Despite identifying this, it was found that backend commonly blocked frontend, as backend developers did not have enough time to work ahead of schedule. This resulted in frontend developers idling while waiting for the backend to become available and catch up, wasting productivity.
Database performance	All queries involving the MongoDB Atlas database were laggy due to poor database performance. Each request took multiple seconds, making the user experience poor. This also made it difficult to debug efficiently, as requests were slow, increasing wait times. This issue occurred on all database queries, regardless of size and will need to be rectified.
Corrupt test data	When inserting data into the database to test endpoints, corrupt or invalid data was inserted into the database inadvertently. This resulted in exceptions being raised in the frontend and backend. These issues included invalid data structures and invalid references where relations were required. In one instance, a defective test script was used to inject data directly into the database, bypassing endpoints and their validation, which corrupted an entire table. This table had to be dropped and productivity was lost fixing this.

## Things to Try in the Next Sprint

Item	Description	Action Owner
Moving across tech stack where necessary	To rectify the blocking issues identified by the incremental deadlines trial, team members will be encouraged to move across the stack to rectify issues outside of their domain if they are comfortable doing so. This will allow frontend developers to make required changes to the backend without having to wait for a backend developer to become available (resolving the blocking). This practice is enabled by the use of pair programming, which means that developers have an understanding of work across the stack.	Tiger
Weekly lecture call / officer recap	One team member will be assigned to create a weekly lecture thread. They will report and summarise lecture content on Slack to remind the team of deliverable comments and suggestions, so that this information can be easily accessed by the team in one location. Whilst all members will continue to watch the lectures, this will assist in easy access to key points.	Antoinette

Postman Testing	A Postman workspace will be shared so that all test data is consistent. This will prevent the corrupt test data issue from recurring, as all data will be validated by the backend, which will validate the request's data. Furthermore, this will increase productivity by allowing all team members to have easy access to validated tests. This also streamlines the insertion of test data, as previously all members had to manually write test scripts or create their own requests via the frontend to insert data and test endpoints.	Tom
-----------------	---	-----

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