COMP3900 Computer Science Project

Retrospectives B

2021 Term 2

By: Team Placeholder

James Dang z5209597 (Frontend)

John Dao z5258962 (Scrum Master and Backend)

Jake Edwards z5114769 (Frontend)

Edward Gauld z5246767 (Backend)

Jaydon Tse z5214494 (Backend)

All members present for retrospectives meeting (20/07/2021 11:00 AM) and retrospectives compilation (24/07/2021 6:00 PM).

What went well

1. Backend integration before frontend development beginning allowed for greater breathing room
    a. Django enabled quicker development times, facilitating the backend's completion before frontend starting
    b. Expectations for the backend to be completed mid-sprint
    c. Preparations for the current sprint were undertaken at the end of the first sprint allowing minimal functionality to be already completed prior to starting the next sprint
2. Direct specifications were created for API endpoints allowed for more effective development
    a. A better understanding of new functionality and how to structure requests to the backend decreased the barrier for development, allowing for features to be implemented quicker
3. Busyness delta project allowed for better utilisation within the backend team
    a. The project allowed for a reduction of redundancy within human resources and expanded the capabilities of the backend team

What didn't go well

1. Human resource management could be improved to enable more even utilisation between teams
    a. In retrospect, a team composition of 3 frontend and 2 backend developers would have served better in increasing member utilisation
2. Server dependency reduced efficiency during frontend development
    a. Backend having to be constantly "pulled" from Github created issues in backend features not being completely functional, requiring troubleshooting
    b. Server runtime dependencies created more barriers to development with developers having to startup local servers to start development
    c. In retrospect, utilising serverless functions (e.g AWS Lambda) would have greatly improved the development process by removing another barrier to development
3. Messaging did not get completed and had to be deployed into the next sprint
    a. Overestimating the complete workload for features overextended the scope of the sprint
4. Backend documentation was not perfect and functionality was not always intuitive
    a. Issues such as having to append a "/" to the end of requests and issues of the like further created delays stemming from troubleshooting

How things that were tried went

Actions' success was determined by a rating of 3 that implied their effectiveness;

Success, Neutral, Failure

Action	**Comment**
Creating specified documentation for new API endpoints as well as more frequent updates to API documentation	Success. Enabled the frontend team to more quickly understand new endpoints
Coordinate more team programming sessions	Neutral. There was not a large need for group programming sessions as development was smoother overall. Main issues stemmed from the scope of sprints
More thorough backend inductions	Neutral. New endpoints and their inductions were already done in brief prior to coming into the current sprint.
Update user stories and backlog structure to more accurately reflect changes in backend specifications	Success. Although scope did not change drastically, changing user stories allowed the team to more effectively develop and put forward better expectations for features.
Increasing frontend team development support when integrating the frontend with the backend	Failure. More resources could have been dedicated in retrospect, especially with the realisation of the overload in the last sprint. As a result, user stories had to be delayed into the next sprint.
Clear planning of agenda for next days of the development process including specific time deadlines for feature implementation	Neutral. Planning allowed for better feature development flow from backend to frontend however, more careful planning should take place in the scope of features.
More communication for frontend requirements to the backend to increase the efficiency of implementation	Success. Requirements from the frontend were relayed to the backend and quickly implemented, enabling the frontend to continue development.
Utilise postman to test backend API more directly	Success. Endpoints were tested with postman and facilitated a better understanding of backend endpoints,
Coordination to complete technical sprint AT LEAST 1-2 days before lab demonstration	Success. Coordination was done 2 days before the sprint, allowing for a complete demonstration to be delivered.

What to try

Action	Assignee
The transition of a backend team member into the frontend team to directly integrate features	John
Integration of Busyness and the delta project into backend functionality	Edward
Facilitating the Busyness Deltra integration and ensuring that integration is done appropriately	Jaydon
Creation of a frontend communication group, specified for frontend team	John
Direct development of delayed features (messaging) and ensuring completion in time for the final demonstration	James
Frontend induction for transitioning backend team member	Jake