COMP3900 Computer Science Project

Retrospectives A

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 (29/06/2021 11:00 AM) and retrospectives compilation (04/07/2021 6:00 PM).

What went well

1. Frontend integration was undertaken with little issues.
    a. The frontend dedicated team had many small meetings.
    b. Frontend inductions and teachings were held very well with work being gradually assigned to individuals who were not as experienced in ReactJS.
2. Django, though hard to grasp, allowed the backend team to quickly integrate new requirements.
    a. Django, Django Rest Framework and its encompassing packages were very modular in design, enabling for quicker "plug and play" development.
    b. Available packages were numerous meaning that more mundane and repetitive work could be done quicker.
3. Team coding sessions worked extremely well and increased efficiency during the development process.
    a. A quicker and higher level of communication above group messages enabled for features to quickly be debugged and integrated.

What didn't go well

1. Backend inductions were not very informative or effective.
    a. Django has a steep learning curve for those who have never learnt it before. This was not anticipated and resulted in an induction which was inadequate in enabling all backend team members to efficiently develop.
    b. This lead to a delay in backend feature release which then resulted in backend functions not being fully integrated by deadlines.
2. Frontend work integration and human resource management.
    a. Frontend had much more work than first anticipated, meaning that the backend development members were left underutilised in comparison to the frontend.
    b. This also resulted in issues with time management, with user stories taking longer than first anticipated and story points not reflecting the true work to be completed.
3. Backend API Documentation.

a. API documentation was often not representative of the actual backend API responses themselves.

b. This is due to the large number of changes that take place during the development process not being quickly updated in the SWAGGER API documentation.

What to try

Action	Assignee
Creating specified documentation for new API endpoints as well as more frequent updates to API documentation	Edward
Coordinate more team programming sessions	John
More thorough backend inductions	Edward
Update user stories and backlog structure to more accurately reflect changes in backend specifications	Jaydon
Increasing frontend team development support when integrating the frontend with the backend	John
Clear planning of agenda for next days of the development process including specific time deadlines for feature implementation	John
More communication for frontend requirements to the backend to increase the efficiency of implementation	James
Utilise postman to test backend API more directly	Jake
Coordination to complete technical sprint AT LEAST 1-2 days before lab demonstration	John