

# Sprint Retro Notes

## Team: Folie a Deux

**Sprint:** Retro for Sprint 4

**Date:** 22/10/2021

**Scrum Master:** Mac Arriero

**Product Owner:** Dale (Tutor)

**Development team:** Mac Arriero, Rohan Poorun, Abdulrahman Ali, Jehan Perera, Aditya Kerhalkar

### **1. Things That Went Well**

Tasks were well-divided based on skills and capabilities of each team member. This included minor fixes as well as deployment of larger functionality, testing (unit and acceptance), and refactoring. Much of the assigned work was simple to do as it consisted only of small edits to the existing codebase.

Communication and teamwork continued to be good between team members, taking place both in and outside of regular meetings.

Since our React knowledge has improved, we have been able to finish the JavaScript-related tasks more effectively.

Also, the teamwork was much more fluid compared to the previous milestones, splitting up into teams of two (pair programming) to work on front end and backend on certain features.

The division of tasks between all group members was more equal than in Sprint 3. Due to improved understanding of the skill and level of expertise of members, we were able to allocate tasks more efficiently. Each task allocated was something that member was able to accomplish.

### **2. Things That Could Have Gone Better**

In some cases, we should have managed our time better to ensure that we did not have to get very large amounts of work done in a single day. Also, we should have been clearer on what is required for a certain task to be 'done' (particularly for non-programming tasks).

### **3. Things That Surprised Us**

When acceptance testing, some modifications had to be made as the acceptance criteria had been written at the beginning of development and some of them did not represent the final product in structure. Also, some features which we thought worked correctly did not end up working and required fixing.

### **4. Lessons Learned**

Removing a purchased book really helped me understand the smaller pieces that make up the microservices.

### **5. Final Thoughts**

#### Things to keep

Keep up the communication on Teams.

Keep up the collaboration between team members on certain issues.