Project Closure
PRCS252 Group J

Reece Drage,
Adam Edwards,
Eugenio Guarino,
Gabryel Mason-Williams

May 2019

Contents

1 How the project went

At the beginning of the project, communication was limited. Communication steadily improved with time as we grew as a team, though issues did arise periodically.

Difficulties with working on the API resulted in lost time and development resources.

The discussion board and the Kanban board provided by GitHub were effectively applied to organise the assigned workload and ease communication between sprint meetings.

Difficulties with CORS resulted in heavily delayed production of a functional web application, though the early discovery of this issue mitigated the loss in time.

As the project went on, we developed an understanding of each others skills, strengths and weaknesses, and attempted to assign tasks based on these characteristics, leading to a more effective Minimum-Viable-Product (MVP) structure. Teamwork and issue management was effective. When issues were found they were assigned in the issues board on GitHub and were communicated and dealt with, where possible.

Code reviews were not always extensive, with some issues being missed due to differences in code structure and ideologies. The clashing ideologies did, at times, produce a break-down in communications.

Code reuse was high, incorporating the reuse of classes and functions with uses through multiple functionalities wherever possible in the Java applications. This resulted in us having a functional product at the end of the project lifespan, which could be improved with limited modification.

2 What would have been done differently

It would have been advantageous to have gathered more feedback, both from end-users and the product owner. This would have enabled us to have a better vision for the end-product, as well as in the planning of a more effective MVP.

Planning a full series of sprints beforehand would have aided with foresight and the overall vision of the product. It would have allowed us to avoid the assignment of dependent tasks as well as with predicting the workload of more difficult user stories.

While group members actively supported in the development of every resultant application, it would have been more effective for a development pipeline to be produced. Individual group members would then be able to develop to their strengths.

3 Lessons learned

In the early stages our understanding of Agile development was lacking because we were unaware of what we should know to implement Agile development appropriately, but our ability to use Agile development methodologies greatly improved as the project developed.

We developed our understanding of Model-View-Controller (MVC) design on larger-scale, multi-dimensional projects. Towards the end of the project, we learned to implement a global MVC structure with the front-end applications, the API and the database, as well as local MVC structures within the individual applications.

We learned more about Restful API development and utilisation. We developed our understanding of Restful API interactions with front-end applications and database services. As well as the development of our own API, we gained an understanding of implementation of other APIs from external sources.

Before the start of the project, we had a limited understanding of Android development, which has greatly improved through the continued development of the mobile application.

Security was not a factor in previous projects and was a new consideration for the development of this project. Over time we developed our understanding of a security-focused programming philosophy.

With previous group assignments that had not performed optimally, communication was lacking. The group learned from this and endeavoured to increase communication and consideration for other group members unique needs and requirements. This resulted in greatly improved team interactions.

Commenting of code in previous projects was less important, as individual group members could easily understand their own code. We learned to comment code in a way that increased readability for other group members, allowing them to understand it fully.

We matured a better understanding of functional and asynchronous JavaScript,

having not developed in JavaScript prior to this project. We learned how to make use of the promise system for asynchronous communication with the API.

Over the course of the project, we developed as a team, learning to work effectively and delegate tasks according to our relative strengths and weaknesses.