32386120 | Alec Ng | 31580129 | Stephen Hu | 54952130 | Tushar Kalra
32136111 | Justin Lane | 90294133 | Yue (Mark) Chen
45909116 | Aviral Garg | 15991152 | Anh Duc (Andrew) Bui

Reflection statement

Things we think we did well and good learning experiences

Splitting up tasks by skill. Having Stephen and Alec take lead on things requiring more technical skill. We ended up in cohesive sub-teams that allowed us to use our skills and also to learn from the more experienced people. Also, having sub teams of people with a similar skill set made it easier to work together, communicate and understand the work that others have done. This was also useful for code reviewing and general input from team members

Coming up with a design that was highly modular. CodePal consisted of the youtube, stackoverflow and code editor modules. This allowed for the majority of development for each module to be done independent of the other modules, making it easier to manage sub teams (since the full team size was 7 people, which is quite large)

Taking ownership of specific features by specific members was a very useful strategy we put into practise. This enabled design decisions to be considered efficiently, and gave the respective owners domain knowledge of their feature, which facilitated our development, as there was always a convenient point of contact for each feature whenever ambiguities arose with debugging, implementation, usage, etc.

Things we could have done differently

Inconsistencies with coding practises. When it came time for integration and the first project demo, we discovered that the code editor team was using a different style to implement html and javascript/jquery than the frontend team. This caused a bit of a headache when it came time to integrate, and involved a fair bit of rewriting code. This could potentially have been avoided if we had simply organized ourselves a bit better, or followed a specific industry standard rather than gung-ho practises.

Taking ownership of deliverables. We often left the organization of deliverables to the team rather than a person, usually ending up with one person taking the lead and having to put in more work than they should have needed to. We probably could have avoided this problem by assigning a person to each milestone as a leader or planning a bit more in advance who should be responsible.

More thorough research in design patterns. The complexity of the front end javascript, especially with the code editor piece could have been reduced with a better choice in design pattern. For example, we chose to use the revealing module and prototype design patterns, but this led to tight coupling between certain submodules that made it hard to test independently and also resulted in a lot of boilerplate code. Using a design pattern like pub-sub would've been more appropriate, given the nature of the interactions between the sub-modules we designed.

32386120 | Alec Ng | 31580129 | Stephen Hu | 54952130 | Tushar Kalra
32136111 | Justin Lane | 90294133 | Yue (Mark) Chen
45909116 | Aviral Garg | 15991152 | Anh Duc (Andrew) Bui

Test-driven development approach. TDD proved to be quite useful in the jpacman-undo assignment. That approach could have been applied in some components of the project. It would have substantially reduced the amount of time spent on debugging and implementation. As is we ended up writing tests after much of the implementation had already been done, causing us to lose some time and efficiency.