Design Project: Reflection Report

Jan Vermeulen

May 16, 2016

1 Personal development plan

- **Time management**: I have learned from previous projects that I start working at a relatively slow pace and take breaks when I feel like it. This means that I put in a lot of hours and then estimated the number of hours that I actually worked productively. For this project I would like to push myself to set clear deadlines and keep a fast pace while working, so less hours are spent to get more work done.
- **Continuous integration**: I would like to learn how to most efficiently use CI/CD tools and automate as much as possible in the development process. As the setup of these systems takes time, they are easily omitted. However, in the long run, this saves a lot of time and effort.

2 Project reflection

2.1 My role

My role in this project consisted mainly of the following tasks:

- Endpoints development using the java spring framework.
- Development and maintenance of JSON parsers for our backend models.
- API documentation using swagger, for each endpoint (together with Juta).
- Spring configuration, authentication and access control.
- Configuration of CI/CD tools: Github, Jenkins and Sonar.
- Setup and monitoring of the branching strategy.
- Maintaining the models component (together with Juta and Jorsi).
- Endpoint and parser unit tests (together with Juta).
- Configuring the gradle build script.

2.2 Continuous integration and Continuous development

As CI/CD expert for this project, I tried to make sure that the development build was always free of errors. There is nothing more annoying about a group project, than solving the build errors of a teammate while working on a certain feature yourself. That is why I tried to monitor the correct use of the branching strategy as much as possible. When implementing new features that could break the current build and affect other team members, I always advised my team members to start a new branch. When a branch needed merging, I tried to assess the impact of te merge for other team members. The Jenkins slack plugin and automatic deployment (and testing) helped to signal when the development build was faulty. This way, build errors

could be signaled and resolved swiftly. Sonar helped to track down sloppy code which could cause problems in the future.

With some exceptions, I think this worked out very well compared to previous projects over the last few years. Continuous integration tools were never used so extensively as they were for design project. In the long run, this had big advantages.

2.3 Time management

Compared to my bachelor's thesis project, my personal time management for design project was improved considerably. I managed to combine other courses and lessons better with design project, than I did back then.

In the development process, the frontend had to wait for my code to be completed, before they could start implementing corresponding features. When errors occurred, they had to wait for the backend to be debugged.

The time management for the development process was improved a lot. When i started working, I tried to concentrate and work productively. In practice, this meant working at my desk instead of laying in the sofa with my laptop, with roommates around me. For this part, my resolutions were implemented correctly.

For the debugging part however, working productively was more difficult. When an error occurs in the frontend, it always had to be researched if it's a backend or frontend error. In practice, this meant I was busy with something other than design project, when questions started occurring on slack. So, for the frontend development process not to be interrupted, I started researching the error in the backend, while communicating debugging information with the frontend developers. This problem however, is due to the flexible work schedule implemented in our team. The only way the time management for frontend debugging can be improved, is to schedule more hackaton-style meetings with the team.

2.4 Overall conclusion

The intentions I had set up for this project, were executed succesfully. There is however always room for improvement. If I had to do it all over again, I would opt for picking a fixed day of the week to schedule a code session with the whole team. This would improve productivity and time management even more.

Concurrently with the development of this project, I started working on an application using Django. This would have been a far better choice as backend framework. At the start of the second semester we opted for spring, to scratch one of the many risks from our risk list, since most of us had some experience with spring. Django has a small learning curve for people with experience of other web frameworks on one hand and python on the other hand. The main reason for this choice is the support for automated object-relational mapping.