

Final Report

Object Oriented Programming Project

Jari van den Broek, Johannes Hagspiel, Nathan Ordonez, Nick van Riet, Lukas Zimmerhackl, Ardy Zwanenburg | CSE1105 | 11-04-2019

Table of Contents

[General 3](#_Toc4415277)

[The Planning 3](#_Toc4415278)

[The Collaboration 3](#_Toc4415279)

[The Communication 3](#_Toc4415280)

[Using gitlab 3](#_Toc4415281)

[What did we learn as a group 3](#_Toc4415282)

[Design decisions 3](#_Toc4415283)

[Work division 3](#_Toc4415284)

[Technological Decisions 3](#_Toc4415285)

[Point for improvement 3](#_Toc4415286)

[Software improvement 3](#_Toc4415287)

[Process Improvement 3](#_Toc4415288)

[Course improvement 3](#_Toc4415289)

[Individual Feedback 4](#_Toc4415290)

[Jari van den Broek 4](#_Toc4415291)

[Strong points 4](#_Toc4415292)

[Weak points 4](#_Toc4415293)

[Conflicts 4](#_Toc4415294)

[Johannes Hagspiel 4](#_Toc4415295)

[Strong points 4](#_Toc4415296)

[Weak points 4](#_Toc4415297)

[Conflicts 4](#_Toc4415298)

[Nathan ordonez 4](#_Toc4415299)

[Strong points 4](#_Toc4415300)

[Weak points 4](#_Toc4415301)

[Conflicts 4](#_Toc4415302)

[Nick van Riet 4](#_Toc4415303)

[Strong points 4](#_Toc4415304)

[Weak points 4](#_Toc4415305)

[Conflicts 5](#_Toc4415306)

[Lukas Zimmerhackl 5](#_Toc4415307)

[Strong points 5](#_Toc4415308)

[Weak points 5](#_Toc4415309)

[Conflicts 5](#_Toc4415310)

[Ardy Zwanenburg 5](#_Toc4415311)

[Strong points 5](#_Toc4415312)

[Weak points 5](#_Toc4415313)

[Conflicts 5](#_Toc4415314)

# General

## The Planning

## The Collaboration

## The Communication

## Using gitlab

## What did we learn as a group

# Design decisions

## Work division

## Technological Decisions

### The Server

In the development journey of our project, we implemented the server in two different ways: once based on telnet and then based on the SPRING framework[[1]](#footnote-1). Even though we achieved great results with our earlier version, our TA pointed out that we had to implement a REST API[[2]](#footnote-2), which was not possible while using telnet. Thus, we had to look for a second solution, where we landed on the SPRING framework. Making use of the tools provided by various SPRING projects such as SPRING Web MVC, we created a REST API. For the servlet container, we used Apache Tomcat and the underlying database was a MYSQL database. Additionally, we used hibernate for object-relational mapping[[3]](#footnote-3). The reason for these design decisions is that they were made following an in-depth tutorial that explained in detail how to best make use of the SPRING framework in the context of an REST API. Thus, what spoke for making use of these tools is that they were accompanied by a guide, that clearly laid out the strengths and the weaknesses of these tools as well as how to best implement them. Obviously, other tools were also considered, such as using a Postgres database. However, after researching the general consensus in the programming community as to the usefulness of all these tools and as well as after having taken into consideration the fact that already a working prototype was created, the decision was made to stick with these services. Looking back, the group unanimously would reaffirm this choice.

### Client Server Communication

As stated previously, based on the TAs feedback, client server communication had to be implemented via a REST API and via JSON[[4]](#footnote-4). The endpoints of the API were set up using the Web section of the SPRING framework and the parsing of POJO’s[[5]](#footnote-5) into JSON and back was achieved through the Jackson. In detail, the client would send, depending on the goal to be achieved, a PUT, GET, POST or DELETE request to a specific URL where the server then accepts this particular request, reads the body and then performs the appropriate action. Again, similarly to the server, this process was implemented by following a tutorial for the REST API and again, other alternatives were taken into consideration. However, after having seen that this particular way of implementing a REST API seems to be the standard, we decided to stick to it. Looking back at this decision, it is clear that this particular implementation provided us with the flexibility to add new and change already existing request endpoints without a lot of trouble. Conclusively, we no doubt would go down this path again.

# Point for improvement

## Software improvement

## Process Improvement

## Course improvement

# Individual Feedback

## Jari van den Broek

### Strong points

### Weak points

### Conflicts

## Johannes Hagspiel

### Strong points

Overall, I did learn a lot about Java programming, mainly about the SPRING framework. I think my stronger points were that I worked independently and on a self-motivated basis, which resulted in me creating on my own the server and the part of the client that is responsible for the communication with the server.

### Weak points

I think one of my weaker points was that I was not really able to explain how the server and the client-server communication worked. I feel particularly disappointed about this as I anticipated this and even told my team mates about it – I mentioned it as a weakness of mine in the README. Additionally, I did not really lead our team and push it during the periods when not much progress was being made, which is was also partially caused by the point I mentioned earlier, my inability to teach other people about things that I have learned.

### Conflicts

As far as I remember, there were no major conflicts in the team. Even though on the surface, this looks good, I do think that this is also a sign of the fact that we did not engage too deeply into the areas that other people were responsible for. Overall, I think I had great team mates, I learned a lot about the SPRING framework and the importance as well as the difficulty of project management were highlighted again to me.

## Nathan ordonez

### Strong points

### Weak points

### Conflicts

## Nick van Riet

### Strong points

### Weak points

### Conflicts

## Lukas Zimmerhackl

### Strong points

### Weak points

### Conflicts

## Ardy Zwanenburg

### Strong points

### Weak points

### Conflicts

1. The Spring Framework is an application framework and inversion of control container for the Java platform. The framework’s core features can be used by any Java application’ - https://en.wikipedia.org/wiki/Spring\_Framework [↑](#footnote-ref-1)
2. ‘Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services ‘ -https://en.wikipedia.org/wiki/Representational\_state\_transfer [↑](#footnote-ref-2)
3. Object-relational mapping is a programming technique that is used to convert data between incompatible type systems using object-oriented programming. [↑](#footnote-ref-3)
4. Java Script Object Notation. [↑](#footnote-ref-4)
5. Plain Old Java Object. [↑](#footnote-ref-5)