OOPP final report

Group 70

April 11th 2019

Nadyne Aretz (4649087)

Matthijs Arnoldus (4928091)

Roderick de Britto Heemskerk (4961269)

Jesse Nieland (4880722)

Roman Sirokov (4769740)

**Index**

Introduction 3

Product 4

Set-up 4

Server and database 4

Graphical user interface 5

CO2 API 5

Gamification 5

Process 7

Reflection 8

Individual feedback 9

Value sensitive design 10

Introduction

For the past seven weeks we worked on making a Java application to encourage people to become more aware of how to be more co2-friendly and to track their progress. To be able to track the progress of the co2-reducing, we implemented several features as tracking vegetarian meals, use of public transportation, solar panel use etc. To encourage people to be more active with tracking their reduction, we wanted to make use of a gamification system. Later in this report we will go more in depth about this system. In this report we will discuss our product, the process while making the application, the reflection on our work, individual feedback, value sensitive design.

In the paragraph surrounding our product, we will briefly discuss what kind of design choices we made. For example what kind of libraries we used and why, what kind of project structure we used, but also what kind of architectural design choices we made.

In the part about the reflection of our work we will briefly discuss what kind of problems occurred during the whole process and how we dealt with those problems. Also we will talk about what maybe could have done better design wise, but also schedule wise.

We will also slightly touch upon our individual process. In this part we will reflect individually on our contributions and what our stronger and weaker points were during the project. In this last part we will also state our initial personal goals and stronger and weaker points and reflect if we achieved our goals and if our view of our stronger and weaker points changed.

Product

Set-up

Before we started with the project, we wanted to form a goal that the whole team agreed on. We wanted to make an application that makes people more aware of being more eco-friendly for the world. We decided that the people for whom we wanted to make the app, were people like friends and family so for the general population.

The process of making the application started by agreeing that we were going to use the view/model/control system that we learned about in the first quarter with OOP. Secondly, we started to think about some extra features and slowly there started to form an idea and structure before our eyes. We cut our project up in five parts: server, database, graphical user interface, co2 API and fractal tree. We divided our roles in the team mainly over the different parts of the project that should we would be working on.

After that, we started working on setting up our project with git and assuring that all of the programs that we were going to use work on everyone’s computers. We made a scrum board, and we looked into the rubric for what we could get points for and how we could fulfill those tasks. We set up also a CI with three different stages. We made a build stage to check if all our code would compile, we made a validate stage wherein the tests run and check if everything works how it should work and lastly we made a checkstyle stage to check that we did not have any checkstyle warnings. We used the gitlab CI, because after some research it came out as the best option, for all because there was much documentation about the gitlab CI.

Server and database

We first chose to work with firebase to use their server-service and to use it for our database. We mainly chose firebase for its scalability, that it is free and online, security and that it provides a wide range of extremely valuable services. Besides that firebase provides database services, it also provides for example authentication services, A/B Tests infrastructure, notification systems, audience analytics, technical (crash) analytics, remote configs. Those are all very impactful tools for improving our product and the user experience. As for developers, it is beneficial if used right. A developer can get profound insight into their product, how it is used, by whom, does it work properly, how to improve it and other things.

Although scalability does not matter for this project, we still saw it as a pro, because this project should give a taste of the ‘real life’ experience and therefore we chose that scalability is also very important. Plus now we would not have to worry anything about scalability, because of the large margin we have.

Because it is online, we also did not have to worry about hosting a database, and we could track all valuable data in the developer console. Also, we use their third-party authentication services. We think security is critical and all the services firebase provides are ensured with high-quality security standards. All requests are handled in HTTPS protocol, thus encrypted. There is also a system of tokens that allow verified sessions.

Some things that could be considered as cons are that the database only accepts NoSQL and that there is relatively low support of Desktop Java Applications. So if there is a personal preference for another database language, or the way of usage requires relational databases, then Firebase is not a good choice. Although we mainly focused on relational databases last quarter with web and database technology, NoSQL turned out to be good enough for us. The low support of Desktop Java Applications was a bit unfortunate, but for us, it did not weight up against all the pros. The problem is that firebase focuses greatly on mobile devices; both android and ios. Although android apps are written on Java, this java differs significantly from desktop version (or more precisely, the platforms differ from each other). Thus, a lot of libraries are not accessible to our platform. We could still fulfill all our needs, but with some workarounds and extra work.

In the end, firebase did not work out for us for the server-side, it was not that it was not compatible with our project, but we were not allowed to use it anymore. More will be explained about this in the paragraph about how the process went. We still kept firebase for the authentication services and as database. Unfortunately, this problem occurred in the fourth week of the project and due to this a lot of our work was lost. So in a very short time, we needed to come up with a new solution for our server. After deliberation with the TA and our team, we decided to use Spring to write our server. Spring is a framework that helps write a server. As a team, we did not know a lot of Spring, but many other groups used it for their project and were very content with it. We had some problems with integrating the Spring server in the rest of our project, but in the end, it all worked out for us. We did not come across any other problems.

Graphical user interface

After we made our decisions how we were going to make the server and which database we were going to use, we needed to choose a graphical user interface that best fitted our project. During the lectures, there were a few examples of GUI’s and some pros and cons for them. The lecturer recommended JavaFX, so we first focused our attention on this GUI. After some research and deliberation with our team and the TA, we decided to use this GUI.

We started working with scene builder which is an interface for working with JavaFX and makes it much more convenient to work with JavaFX. After some practicing, we started working on making the views for all the different features that we were planning on implementing.

CO2 API

In our research for an API, which could help us calculate the reduction of carbon emission, we came across some problems. First of all, it was quite hard to find a good API, which we could use for as many features as possible. The OOPP team had recommended a particular site, but the main problem that we had with that site was that it could not calculate the reduction of carbon emission for a few features that we wanted to implement. Also, we found that too much information needed to be filled in for the calculations. This would be too much of an inconvenience for the users of our application.

Most of the API’s on the internet for calculating the carbon emission reduction were too basic, or we needed a license to use it. Therefore it took us some time until we found two reasonable API’s. However, then occurred another problem. We tried to contact one of the API holders, but even after two weeks we still had no response. After some discussing with our TA, we decided that it would be best to contact another organization. We contacted the organization CoolClimate to ask if we could use their API, and they responded quite quickly. After that, it went quite smoothly, and we were able to use the API for our project. We incorporated the API in our project by using the data from the API for calculating the carbon emissions of cars and public transport per kilometer, in order to be able to calculate the reduction of emissions when cycling or taking public transport instead of driving by car.

Gamification

One of the requirements or the project was to make use of gamification. We as a team got a lot of freedom how we wanted to do this and had immediately many ideas. Our main idea was to make use of scores. For every feature, a user can get scores. So if they thrive with one feature they have a high score for that one, if they do bad for another feature, the score will be very low.

We wanted to make the progress of a user more visual, so we implemented a score bar and a fractal tree. The fractal tree will get bigger when the scores get bigger. Also, different colored branches are based on different features. This makes it easier for the user to see for which features they thrive and for which not.

Lastly, we implemented also achievement badges. We made seven batches: an Eskimo badge, a no-car badge, a muscle badge, first points badge, vegan badge, 100th level badge and a public transport badge.

Process

Reflection

Individual feedback

Value sensitive design