OOPP Final Report

OOPP Group 40

April 4, 2019

Contents

List of Figures

Preface

Group members:

- Andreas Achilleos
- Alp Capanoglu
- Jordy Koemans
- Omar Sheasha
- Sung Park
- Martijn Staal (4790162)
- Just van der Veeken

Chapter 1

Product

1.1 Description and goal

Our product is called *Picle*, which is an acronym for "People Interactively Creating a Livable Environment". Picle aims to enable users to make their ecological footprint smaller by tracking the saved amount of CO_2-equivalents, comparing them with their friends and rewarding them achievements.

1.2 Product structure

The product consists of a client, an application server and a database server.

The application server and the client communicate with each other via HTTP Basic Authenticated requests in a RESTful API. This API is defined in the `doc/` folder of this project.

Picle is focused around *activities*. Any thing that a user can do that has an impact on their CO_2 emissions could be made into an activity. We however needed to make a compromise to keep the score calculation maintainable and user friendly. For more accuracy you need more parameters, but the user obviously wants to fill in the least possible things.

Activities are added to their personal database of activities by the user. Each activity modifies the score, which is the amount of kilograms of saved CO_2-equivalent emissions saved, of the user. For each activity, we've established a formula to calculate this score with either one or two integer parameters. These formulas are obviously an approximation of reality, but with our research we have confidence that it portrays reality accurately enough for the goals of our application.

1.2.1 Client

The client is "thin", which is to say that the client doesn't store or calculate anything. This is to protect the integrity of the data and scores so the attack surface for possible cheaters is minimized. The thin nature of the client also enables extreme portability: users can log in on any device and still have all their data ready. The downside of this, is that a lot of data has to be fetched from the server when a new client connect. Another possible issue is that a user is not able to view their activities and score when their client is not connected to the server.

The client has a graphical user interface, which is made using JavaFX.[1]

---

[1] https://openjfx.io/

1.2.2 Application server

The application server is a server written in Java 11 using the Spring Boot framework.[2] The application server is the computational brain of the whole Picle application.

When a user adds an activity, the client will send a `POST` request to the server. The server then first checks if the user is authenticated correctly. If it is authenticated correctly, the server calculates the score and stores the activity in the database.

Most other workflow on the server works in a comparable way: an authenticated HTTP request is sent by the client which is then processed accordingly by the application server.

This server centric approach makes it easy to have full control over the Picle application, but it has some downsides. Scaling the Picle application for a big amount of users becomes relatively hard with this approach, certainly when a client always retrieves all their data when a user starts the client.

1.2.3 Database server

For the database server, we have chosen to use MariaDB.[3] We chose MariaDB because it is a simple, free and open source SQL server.

1.3 Security

While designing and building our product we have mostly looked at three facets of security: data integrity, authentication and confidentiality of user data.

1.3.1 Data integrity

Since Picle is also partially a game, with the competitive aspects of comparing scores, it has to be a fair game. Therefore, we need to make sure that cheaters have no chance. This is done by making the server trust the client for absolutely the minimum amount necessary. Everything that can be done on the server, is done on the server.

1.3.2 Authentication

Every individual request from the client to the server is authenticated using HTTP Basic Authentication. With this approach, there is no such thing as a 'session', so any session stealing kind of attack is impossible. The passwords are stored hashed and salted on the database using PBKDF2.

1.3.3 Confidentiality of user data

Our application server is designed and implemented in such a way that a user can only access their own activities. This is important, because a user's activities may say a lot about their habits, if they use Picle a lot.

---

[2]https://spring.io/projects/spring-boot
[3]https://mariadb.org/

Chapter 2

Process

Along the path of this project, we adhered to the Scrum workflow and decided upon splitting the team into a client-side and server-side team. We split up into two teams so that we can work efficiently, and focus on implementing either side of Picle to be able to have a clear and better understanding one specific side.

2.1 Scrum

Scrum workflow is an agile way of managing a project. It entails having a Scrum Board which contains all the tasks and goals we need to accomplish that are stored in our backlog. The Scrum Board is maintained on GitLab's issue board on a consistent basis along with moving our tasks into their appropriate board whether it is in progress, being tested or completed.

2.1.1 Meetings

During our weekly meetings, we discussed issues and tasks we needed to tackle and achieve to further develop Picle. The designated chair of the meeting and the person taking down minutes was changed per meeting so that everyone got the experience of both roles.

During that meeting, we assigned tasks to each member so that their roles are clear throughout the upcoming sprint. That was necessary so that we could all concurrently work on different aspects of Picle in order to be more time and work efficient. Adhering to a modular Model-View-Controller design pattern enabled us to work concurrently.

In addition, we discussed any decisions we need to conclude upon like which API to use or which DBMS to use. After having all tasks and decisions made clear, we started coding throughout the specified duration of each sprint. Throughout a sprint, we made sure to update our Scrum Board, as we worked on tasks, and that we offered help to members of the team. Moreover, we made sure that we finished all our work during the specified sprint in order to not fall behind and be capable of meeting the deadlines for our demos.

After each sprint, we took some time in the meeting to review our progress during that sprint, what challenges we might have encountered and how we overcame them, as well as present the demo to our TA. We also push to GitLab screenshots of our test coverage, Checkstyle report, Scrum Board, release with a tag, sprint plan, and sprint review.

Chapter 3

Reflection

3.1 General Overview

The course provided the groups with a unique experience of a real-world developing scenario. We had to adapt to the difficulties we faced while trying to apply most all of the knowledge weve acquired so far in the program. Our result is definitely not perfect but each member is now familiar with the process of creating an actual product.

3.2 Group Management

Each member of the group was present in every weekly lab meeting. The group was split into two teams each one working on client and server individually.

We managed our weekly tasks for each sprint through the SCRUM board on GitLab. Throughout the sprint we updated our backlog and assigned individual tasks to team members and tracked the progress through WIP merge requests so we could give feedback and discuss on additions and changes.

3.2.1 Sprints

Our sprint planning was off, especially during the first sprint. Our expectations were quite often higher than expected so we ended up spending more time troubleshooting new features. This meant that our Friday goal of merging new features was not met and we spent the weekend finalizing our changes.

Reflecting on our planning, we could have set more strict goals for each sprint aiming for a more stable timetable.

3.2.2 Communication

The group was having a weekly brief on every meeting with the our TA being present. This is the moment of the week where each member got updated on what everyone was working on and what is the status of ongoing implementations and features.

Throughout the week each member had its own planning with just the sprint deadline to look at. Updating the team during the entire week would provide more insight on the projects concept and/or signal the rest of the members if the workload needs to be shifted for a particular implementation.

3.3 Developing Picle

The product covers all basic features listed in the courses rubric. The client itself is pretty lightweight as most of the calculations are done on the server side. More features are always welcome and a project can always expand ofcourse.

3.3.1 The Client

Improvements for the client would include, cleaner animated user interface components, more configuration options, such as customizing your display, and additional gamification aspects that provide even more stimuli to the user than what is already available.

3.3.2 The Server

One of the tradeoffs we had to accept was the accuracy of our score calculation. For the sake of simplicity and creating a user-friendly design we had to limit our activity parameters leading to a more inaccurate result. Additional options for individual activities would greatly increase the accuracy of carbon savings and provide more detailed information.

3.4 Following the OOP Project

The structure of the course helped both experienced and inexperienced team members to get familiarized with more professional developing environment. The basics of Git and team planning are key-skills for any upcoming computer scientist.

3.4.1 Real-World Scenario

To recreate a customer-developer relationship the groups were mostly left in the dark on any technical matters regarding the project. The lack of additional help dragged the group progress quite down. Developing new features proved rather slow in the beginning.

An improvement from the course staff would be to provide basic information on a more complete toolset that a developer can use to kickstart a project.

Chapter 4

Individual feedback

4.1 Andreas Achilleos

Kickstarting the project was especially difficult. The beginning was slow and our progress wasnt really visible. Things started going smoothly once we had built a proper infrastructure for the project and had our basic functionalities nailed down. By sprint 2 everything went by a lot quicker. Each member was assigned on a different part of the project development while keeping our client server team division.

I worked almost exclusively on the User Interface; designed the initial wireframes and then continued to implement the scenes using JavaFX. I used CSS files to style the components and, added custom libraries like JFoenix and AnimateFX to provide extra functionality. I designed and animated our product logo and used it throughout the project. When the basic scenes of the client were built, the rest of the client team members and I wrote controllers to deal with user input and error handling. Then I connected everything together to create the finalized user interface.

Getting accustomed with the new tools that were introduced to me was the most difficult part of the development process and the part I had most trouble with. Having no prior experience in team-developing and large scale projects, learning how to use Git was one of the most crucial tasks. I spent many hours researching unknown software and terminologies so I could get on the same level as some of my more experienced teammates.

To compensate for most of my weaknesses, I was present in all group meetings and met all deadlines both from our team and the course schedule. Some features took longer to implement and the more experienced team members had to work extra to fix errors that I was generating. Getting feedback from different members helped me personally to understand complex topics and generate a result that we were all happy with.

Overall I am happy with our team performance and our final product. No real problems were raised that we couldnt resolve within our team. Everyone contributed to produce a very satisfying result.

4.2 Alp Capanoglu

I am satisfied with my contributions to the project, and I surpassed the expectations I set upon myself. I was able to work with the group in a productive manner and feel that I have communicated efficiently with my teammates, I have also contributed and learned more than I

had thought.

I feel like my strongest part was how I was able to make up what knowledge I lacked in the first weeks and contribute meaningfully to the project. I met all my self-declared goals on time, wrote many of the tests and most of server-side authentication as well as some work on the other layers of the server. My overall knowledge on the server became evident to myself when in a meeting we wanted to add a new extension to one of our server endpoints, and I was able to write the necessary code on all layers that Sung set up, and test it in a small amount of time. I am relieved and proud, because my biggest doubt of myself was that I would not be able to learn time management skills, or underperform.

Even though I have met all my deadlines, I feel I used my time inefficiently, as my time working on the project was intermittent and usually in late hours. Even though compliant with the structure and spirit of Scrum, this has proven to be inefficient in every other sense than producing code. Now that I realize that this is bad practice, I aim to improve my time management skills even further; this would also what I would change were we to hypothetically repeat this project.

The most difficulty I had in the project was the first few weeks, where I had to learn how to use project management and building software as well as version control tools. I also had to learn about various methods of testing, and how filters and beans work as defined in Spring. This I overcame through consistent and hard work. I also asked friends, primarily Martijn and Sung, when I needed help of any sort. Ironically this period was perhaps the one in which I best managed my time (on a personal scale). The end result of passing this hurdle however has been the large amount of new skills I have developed and meaningful contributions I have made to the project.

4.3   Jordy Koemans

For this project I mostly worked on the client-side of our team. At first I was going to be part of coding the features but this didn't go so well. I was having difficulty with getting started and keeping up with the speed of the project. It was pointed out that I wasn't clear on what I was working on and what I was having trouble with which I agreed with. Clear communication in group projects is something that I have difficulty with and was something I wanted to work on. After that was pointed out I tried focussing on it more which I think I did improve on.

After that I started working on researching the activities which was able to do and it helped me contribute more to the project which had been a problem for me and the group up to that point. Once that was done I started of increasing test coverage which proved to be a difficult task for me but thanks to a lot of help from the team members I was able to do that succesfully.

In the end I did improve on the points I wanted to work on but not as much as I'd have liked. I lost track of the project from the start and my communication wasn't clear enough, the last few weeks went a lot better in that regard and it did help me a lot. I would like to get better at coding in Java so I can contribute more in following projects but looking at the work my team members made did teach me a lot.

4.4   Omar Sheasha

Throughout this project, I have been working on the client-side aspect of PICLE. I have learnt how to coherently cooperate in a team and take on tasks that I did not have experience in. I learnt how to manage my time and adapt my schedule to always meet the deadline for the demo, and keep the teams workflow steady without holding them back. I also mostly learnt how

all the different components play a role in reaching our goal. I have been able to take on the responsibility of being a chair of our weekly meetings as well as taking minutes in our meetings. Being the chair of the meeting required me to be able to identify our upcoming challenges, and clearly present them to the team so that we can decide upon how to tackle them and proceed onwards. I believe I was able to present myself well, make those points clear and take everyones opinion on how to tackle the discussed issues. I also learnt to take criticism or feedback and act on it as well as not shunning away from reaching out for help when I am faced with something I can not resolve. I believe I have acted upon all the points of my personal development plan.

4.5   Sung Park

I worked on the server-side part of out team. My main contribution was structuring the server side by splitting the server into a N-tier layer where each layer has a smaller scope of function. This structure helped organizing, knowing where to implement what, and testing much easier. Then implementing the functionalities of activities, users, and scores with tests were part of my job as well.

My strongest point for this project was finishing the works assigned to myself on time without any trouble. At the end of the meeting, I would know what I would have to do before the next meeting and I would have that task done before the meeting next week. Since my task was mostly implementing server side implementations, I would also have to keep the test coverage up to par whenever I add a functionality, which I did not let down on before the merge request was ready to merge. I wouldn't leave any mess on our master branch.

One of my weaker point was communication. At the beginning, I worked on structuring the server side, which of course needs to be communicated to the server side people of our group. However, I was not clear to them with that. It is partly because I myself was not sure with the structure until I was able to understand them which was quite late. Gradually, I communicated even more using WhatsApp, but I found GitLab really useful for communication, code review and discussion, which at the end our whole team communicated very well including me.

One small problem I had was not reviewing every code thoroughly. One time, I had implemented a method which was already implemented. This was because I had not reviewed the merge request which implemented that method. It caused a bit of confusion amongst the server-side people, but through communication and code review from GitLab this was solved and pointed out easily.

4.6   Martijn Staal

I am very satisfied with what I contributed to the project, on all parts: code, organisation, planning and with the report. I also think that I helped my team members with their tasks and with with learning things I am more experienced with (mostly git, LaTeX). Although I think that my direct way of communication might have seemed a bit harsh in the first weeks. But I think that all in all I can positively say that I have improved my collaboration skills, as I intended in my Personal Development Plan.

My weaker point during this project was that I didn't do that much research into the framework we used for building the application. I mostly asked Alp how everything worked and used the existing code in the project as examples. This however does show that we were able to effectively work in a team.

I think that there were little problems during the project in our group. Most of the time we were able to find consensus on the issues we faced, which were mostly design choices. Sometimes

8

it took more than one meeting to make a decision, but I think that this was necessary. In the end, this gave us the extra time we needed to make a final decision.

4.7   Just van der Veeken

As with every project, teamwork makes the dream work, but that is easier said than done.

Personally I think the project was a successful team effort in which every team member was satisfied with their assigned tasks and the final product while maintaining a balanced workload.

I experienced it to be very educational in terms of tackling challenges. It really thought me how to properly research technology I did not know which I imagine to be extremely useful in the future.

For this project I feel like my contribution was satisfactory. I think I can safely say I have at least some knowledge about every aspect of the project. This put me in a position where I was able to help out my team members often, which matches the ambitions I had at the start of the project. Also, I tried to communicate with the others frequently about their progress in order to spot bottlenecks for the deadlines we had to make. Addressing these to the team sometimes created the opportunity to discuss possible solutions in advance, such that escalation of upcoming problems was prevented. Besides that I tried keeping an eye on how everyone was feeling about the work they were doing. As a result some of us shifted their main focus for the project so that they experienced it to be more pleasant or they felt that they were contributing more that way.

The most difficult aspect of the project was time management. A lack of proper planning resulted in me learning it the hard way. Namely, working all night long to pass one of the deadlines. Complementary to planning, using your time efficiently is a whole different skill which I could have improved on. For some tasks I spent much time struggling with the implementation while doing some more extensive research in advance could have saved me a lot of time.

In conclusion, I think the project thought me some valuable lessons on project management and team work with proper time management being the hardest nut to crack.

Chapter 5

Value Sensitive Design

Programs have an impact on people. But not only on the people who directly use said programs. There are also other stakeholders. In this chapter we will elaborate on how the design and implementation of Picle would have changed, if we would have looked at more than only the users as stakeholders of Picle. The other stakeholder group we will discuss will be that of (wild) animals.

This stakeholder analysis is important, because we, as programmers and computer scientists, have a moral obligation to take this into account. The specialization of what people do that started with the agricultural revolution and has increased ever since, means that a small group of programmers decide what the non-programmer population can do with their computers. This is a great power, and thus comes with great responsibilities of which we should be all aware.

If we would have focused more on these two stakeholder groups, gamification would have been a bigger part of the final product. This because this drives competition between users, which will in turn help the users be more engaged and have more fun and the users will make their lifestyle more ecological. Which in turn will increase animal welfare.

The two stakeholder groups are thus *users* and *(wild) animals*. The users of our application are an easily definable group: it is just everyone who installs the client and uses Picle. The other group, the (wild) animals, are harder to define. Because how does our application have an effect on this group? The users of our application have an impact on the animals around them: the wild animals that live in their region, but also the animals they eat and those that live in places endangered by climate change or pollution. If Picle encourages our users to live a more earth-friendly life, they also improve the lives of the animals they affect.

We can easily see that the interests of both stakeholder groups overlap a lot: they both benefit from a more earth-friendly lifestyle. However, there is some tension between both groups. For example, if we would introduce a penalty for eating meat or consuming animal products, that would be better for the animal welfare, but not all of our users would probably like to change to a completely vegan diet.

If we focus on the overlapping interests, the most important thing would be to make Picle more accurate and to add more possible activities. For this, there is still a lot of room for improvement. We are obviously no environmental scientists, so that is a group of people we would need a lot of advice for. The more accurate Picle is, the better it can help people with an earth-friendly lifestyle.

For adding more activities, the best way would be to crowd-source ideas from our users. We are with just seven people, so the diversity of the ideas from our users will definitely be greater than what we ourselves could come up with. This would however require a redesign of the GUI.