

GROUP 98 - OOPP Report

1. General information about the project

No big project consisting of 7 people would go well without planning, collaboration, proper job division and communication. In group 98, we dedicated a lot of time to make sure everyone in the group was up to date with current updates/changes to our program, either with weekly meetings and reviews, or concise but clear commit messages.

At the start of every week we would have a meeting to discuss the previous weeks' achievements and encountered challenges. Every member of the team would update the team on what he did that week, what went well, what did not, and any improvements and comments he would have. After that, we would create an agenda for next week, divide our workload and start programming.

During the first weeks of the project, we encountered problems with job division - some people would get a significant portion of the workload while others would only have a chance to contribute less. This affected the morale of some team members, but it was quickly brought up in the next weekly meeting and it was made sure that every member gets their fair share of work.

Another problem that we encountered was miscommunications. There have been a few cases, mostly in the early weeks, where the same task would be done simultaneously by more than one person, making the other person's work obsolete. After one particularly big case of this (one person did 3-4 hours of work for nothing), we made sure that this never happens again and started dividing work very clearly as well as improving real-time communication while working on something.

Our communications were done by a few means:

- * Weekly issues and goals were posted on our scrumboard;
- * Minor issues and quick questions/decisions were discussed in our WhatsApp group;
- * Major decisions, merge requests and big problems were solved during our weekly meetings.

It might be worthy to note that to keep up with the schedule, we not only made use of the mandatory Monday meetings, but organized additional programming meetups in the TU Delft library.

Generally, as the project went on, everybody in their team started to understand what their responsibilities are, and how to approach them best. It might have helped that at the start of the project everybody divided their part of the work (two people for GUI, three people for

server/database, and two people for testing/general problems), and stuck to this schedule through the rest of the project.

It is fair to say that at the end of this project everyone on our team improved in some way. It is obvious that we saw how important planning and communications are in a project of this scale, and, based on how serious problems were encountered less and less as the project went on, it is safe to say that we learned how to work in a team and improved our planning and communications skills.

Moreover, in addition to learning how to use Git, everybody had to program something they have never encountered before. Whether it was server, GUI or database, every member of the team gained important programming knowledge in an area they haven't encountered before

To sum up, this project was very challenging, but at the end very rewarding, since every single one of us learned something new and improved their skills in multiple areas of computer science.

2. Design decisions

One of the major decisions that we made as a team, was that at the start of the group project, we split the work on the application into groups. We decided that Omar, Alex and Vidas will be working on the server-side, Alvaro and Jonathan – on the databases, and Lachezar and Berry will design the GUI....

Concerning the technological choices that the team made, firstly, we researched the frameworks and libraries that we were going to need to build the application. The team then decided to build the server using the Spring boot framework, because of several reasons. First of all our TA had good experiences with it and recommended it to us as well. Furthermore, this framework is very well documented, including step by step guides. This helped us a lot on getting started, and with the build in ObjectMapper class, sending and retrieving JSON objects becomes really easy. On the client side we used the java HttpURLConnection class to build our own handler class, because it's very simple and flexible. Concerning the GUI, we chose JavaFX, because it is easier to implement than Swing and AWT while it offers everything we needed. Furthermore, more than 70 libraries are implemented for JavaFX and thus, it was going to be much easier to design the application. The images were created on Adobe Illustrator, because we decided that we are going to have an unique interface of the application. The server hosting is done on Heroku, we pretty much followed up on the course staff's recommendations there because it's free and has great support for data storage (through pgAdmin for example). The database is hosted on Heroku through pgAdmin, since we have experience with it, and it has integration with Heroku. We preferred to use MongoDB at first, but sadly this was not possible in combination with Heroku.

3. Points for improvement

There is much space for improvement in the software that we designed. For example, the GUI can be made even more advanced, e.g. background animations can be added to the main screen and the quality of the front-end code can be improved.

There is especially much room for improvement regarding testing: by using integration testing, end-to-end testing and TestFX for the GUI - whereas now we only have JUnit 5 tests, and sort-of integration testing on the database.

Apart from points for improvement, there are also things that have improved a lot over the course of the project. We have seen each other grow, both individually and as a team. To name a few things: The usage of git, work ethics, division of work, coding style, overall coding performance and especially communication.

The last bit I want to point out here is the part of knowing what is required part, because this has been and is still our biggest issue in my opinion.

I feel it could have made a big difference had we took a look in the rubrik more often and, sadly, I haven't seen much improvement on this point.

As a whole, the making of the project itself could have been improved by the team members. Although the team synced well after a few weeks, the planning did not go as expected and we had to create a new thoroughly thought plan in the middle of the timespan of the project.

4. Individual feedback

Lachezar Lesichkov

I was responsible for the delivery of the GUI part of the application. Having made several front-end applications before, i was confident that i would best fit in a team, where I design the application. I had no previous experience with Java GUI libraries and frameworks, but that was not a setback at the time.

After the team distributed the work among the members, I was eager to find out how this project is going to work out, because as I stated in my PDP, my strongest motivation was to find out how real applications with a team are built. First weeks were the hardest, because everything that I was supposed to use was new to me. Thus, in spite of having been ahead of schedule in week 1, I was late in weeks 2 and 3. After that, I picked up my pace and managed to always stay ahead of schedule and, if possible, implement extra features.

My strongest point during this point was the dedication towards the project. I stumbled upon many problems, but my motivation to add more features to the application i am building drove me forward. As a weaker point, I would single out the bad planning that i did for weeks 2 and 3, where i was behind schedule. At that point, i had not realized that the most work to be done is at the start. After that was easy.

Concerning my team, I am of the opinion that we managed to sync very well. We all worked very well together and everyone managed to keep up with the established pace. Whenever one had problem with something, others tried and helped him. The only conflict that we had was in the middle of the project and it was about the bad planning and the division of the work. However, we solved it in no time and were on track again.

Bejruz Domínguez Adilova

During the OOP project I was part of the duo in charge of the GUI, together with Lachezar. My work consisted mostly of graphics, styling, and layout. The team functioned surprisingly well, given how all the members were selected randomly. I found that most of the group members were more ambitious with the final grade than I was, but this did not stop me from contributing to the group whenever possible and whenever requested.

Admittedly, my performance during the first two weeks was lackluster. Nevertheless, I picked up my pace shortly after and found the set of skills I could bring to the table that would uniquely contribute towards the final product and the project as a whole. A factor that added towards my less than ideal start was the lack of clear delineation of work, which when brought up to our attention, we resolved swiftly.

Much to my surprise, aside from the work delineation mishap at the start, which was resolved with more abundant and clearer communication, there were no other notable conflict between the team members. The greatest testament to how our group functioned like a well-oiled machine and with an uncanny lack of conflict is that we finished the app two weeks in advance of the deadline, which allowed us to further work on extra features and polish out the existing ones.

In conclusion, my greatest shortcoming during the project was, in my opinion, my lack of ambition at the start, where as I consider my strong point to be my decent communication skills and creativity.

Vidas Bacevičius

For this project, I was responsible for the testing/general help on the project. At first I was tasked to work on the server with Alexander and Omar, but quickly realized that I couldn't pick up on all the new material so quickly, and my work would have to be done by someone else to keep up with the schedule, so I started to stick with what I had experience in and could help with - testing and quality. The first quarter of CSE, the OOP course in particular really helped here, since, during it I learned about unit testing and could apply and expand my knowledge here.

Since I started to work on testing, my contribution to the project increased vastly. I also continued communicating with Alexander, helping him if he needed it, and asking for help and advice myself. I also liked that everyone was very enthusiastic about the project, which increased my morale and made the work on this project more enjoyable.

At the start of the project I had some problems with Git, which stemmed mostly from inexperience with version control system. Thankfully, my teammates helped me when I encountered any issues, for which I am very thankful.

Overall, I really liked this course, because not only did I get to improve my computer science/programming and version control skills, but also see how a project of this scale is executed and how important planning and communications are.

Jonathan Garack

I am satisfied with the progress I did during this quarter while working on the project. I think our whole group worked very efficient. I had a lot of opportunities to improve my programming skills and I think I used each of them. Each time I had a problem my team or our TA were really helpful.

I mainly focused on the Database part of the project. I created and designed the postgresql database, during this time I finally found useful lectures from previous quarters. I also learned how to connect the database part of the project with the application. The most difficult part of my work, was testing, but finally i managed to do them.

Of course I had few problems doing the project, but the communication between each of the team members was really good. Each time I had a problem someone helped me.

At the beginning of the project definitely one of my weak points was that I have never used most programs that the teaching team asked us to use. I had a lot of problems with gitlab and sourcetree, especially source but after few weeks i got used to this and it was fine.

I am happy with the teams dynamic and I hope there will be more projects like this one in the next quarters.

Álvaro Buj Mancha

My job in the project has been mainly focused on the deployment of our server and database in Heroku. On the first weeks Jonathan and I were in charge of the database, we worked on the original schema and on uploading it to Heroku to work with it. After this we started with the deployment of the app itself and later we split the job: Jonathan worked mainly on the database and I worked on the server.

It definitely took me a while to pick up the pace, the first couple of weeks my job was slow because I wasn't understanding the overall structure of our project and I was looking at it from the wrong place and this is where the main problem I had appeared: I was working on deploying the entire app on Heroku and for that I spent a week and a half on solving errors related with the JavaFX Heroku buildpack version so I worked on creating our own JavaFX buildpack. This was a waste of time due to the fact that we had to deploy the server and not the entire app. However, in that time I faced so many Heroku errors that I already knew all the fixes when they appeared later so it wasn't that bad after all.

I didn't work as much as I'd liked with the code: the only work I did with it was with the POM file dependencies and the modifications to the server-side code that I had to do in order to deploy it to Heroku. But this was because the part I was assigned to do didn't require more and even though it wasn't exactly what I wanted I got to learn what I wrote in my personal development plan which was learning how to work on an environment that was similar to what a job would be.

The only problem I had with my team is that when I created 2 submodules to work on the server-side and client-side separately we had issues due to the fact that some classes were needed both in the server and client side. To fix this I moved my work to a new repository and I worked on it on my own to create an independent server-side module.

As a conclusion I would say that I really enjoyed the project and I found the weekly meetings very interesting, seeing our TA Tijmen asking more from us felt as if we were doing an actual job and my teammates took it really professionally, worked really hard and asked a lot from me so I'm really glad I got to work with them.

Omar Hussein

Through out this project I would say I had consistent contribution, I mainly worked on backend for client and server but also helped with other fields. I have completed all the tasks assigned to me on time, and I have also asked for extra work on some occasions. Working with my team was enjoyable as we had a high level of positive communication, which was hard to achieve at the beginning but as we got more accustomed to each others working styles it became easier.

I would say communication was one of our strong points at the end of the project as we'd always work together whenever an issue was raised and it would, usually, be solved quickly. Each team member became used to a specific aspect in the application although we all had general knowledge of its internal working which also sped up the completion of tasks.

I wouldn't say I had any downfalls during the development of the application, but on a few occasions especially during the first 3 weeks of the application if I felt that other team members weren't putting enough work I would slack off myself. This was not the right approach to such a problem, but better communication between us and delegation of work helped offset this issue.

On the other hand, I had established a general idea of how the app should work very early on which made the tasks that had been assigned to me easier to do as I did not need to research that much as I had previous knowledge on most aspects.

Alexander van Vugt

I started the project very motivated and full of energy, but as the end came near my motivation started declining, which is probably mostly because I'm not feeling well overall and important exams closing in, so more of a personal issue rather than project related. This also caused me to fall back into my procrastination habits.

Throughout the project, I believe my stronger points were overview and coordination, as well as dedication, all be it weaker towards the end.

I didn't have major conflicts during the project, at most some "healthy irritation" at some points.

However, most of those cases were easily solved with some settling, coordination or a little talk.

A few weeks into the project, the unbalanced division of work started to spark a conflict, but we quickly talked it down and solved

the issue. Another issue was that not everybody had a clear overview on the repository and the project in general.

An extra meeting was called in, during which we went through the entire repository on a conceptual level.

This helped everyone gain back a clear overview, whilst also making us revise the structure of the repository.

This reviewed structure was very useful in the weeks after that, and there was a more clear separation between modules.

5. Value Sensitive Design

1. How would you define your main concept?

Clima Mutatio Dolus' main concept was to help people reduce the carbon footprint. Our application focuses mostly on what we can do as individual users, it helps them find environment friendly habits and rewards them for reducing their carbon footprint . The main idea behind the concept is to treat our users in a way that they feel accomplished by keeping up the good habits; we do that by keeping track of the points, which allows the players to compete with their friends and receive rewards in the form of achievement medals.

2. How would you define in general terms the stakeholders/values you want to design?

Our app is very user friendly, it is focused on being used as a tracker for our environment related habits and wants to distance itself from what the common social media app does which is generate addiction.

We wanted an app that is simple enough to not have anyone stop using it because of it complexity but informative enough that people can understand the real difference our good habits can make for the world.

3. Which sources would you consult in order to gain theoretical insights into the stakeholders or values you want to design for: academic literature, artworks, experts in domain different than yours...; give some general indication not necessarily titles of books or names of experts: e.g. "social scientists who have study the phenomenon X, documentaries on discrimination of the group Y, legal/philosophical literature on the right to Z, political studies on the meaning of demo etc.)

To accomplish the main goal of the application we as developers needed a lot of sources of knowledge, mainly statistics about the carbon footprint and in which measure we as individuals with our damaging habits are to blame.

When designing this application we tried to focus on this specific knowledge as much as we could, but there is still a lot of work to make the app completely accurate.

As developers we studied all the resources about climate change and CO2 pollution we could get our hands on and that made us realise which human habits are the ones that damage the environment the most. We used this knowledge to find fixes for this issues in the form of habits that are more environment-friendly.

4. Your concept involves the realization of at least two values OR one or more values for at least two different stakeholders

The first thing you can see when you open the app is the login screen where you can see we focused on privacy:

When designing the app we created it so that it requires the minimum of the users vulnerable information:

- A unique username
- A password
- A security question

Note that we don't ask for an email account so the information we have of our users is very superficial and that makes it easier for users that are concerned about their privacy.

All user data is stored in a database and the most vulnerable parts are encoded.

The application also takes care of users privacy by blocking all further friend requests from users you've declined as friends.

Also, we wanted it to be simple and fast , we wanted an app where our users can simply log in, put in their habits and continue with their lives.

We didn't want to create a social network, we wanted people to focus on their habits and the friends part of it is just to add a little bit of competition.

That's why the first thing that appears when logging in is a set of habits that you can add points to, we didn't want to divert the attention of our users from our main goal.