

Calin Georgescu, 4794672

Danila Romanov, 4835786

Dorka Hévizi, 4944658

Francine Biazin do Nascimento, 4840666

Pradhyumnaa Ganapathi Subramanian, 4810317

Yash Kalia, 4785304

OOP Project: Final Report

Group 7

Table of Contents

Product	3
Process	4
Reflection	5
Individual Feedback	6
Calin Georgescu	6
Danila Romanov	6
Dorka Hévizi	7
Francine Biazin do Nascimento	7
Pradhyumnaa Ganapathi Subramanian	8
Yash Kalia	9
Value Sensitive Design	9

Product

The first major decision we had to make regarding the product was to choose a framework that would facilitate the development of its tripartite structure: a server, a client, and a graphical user interface. After some initial research, we came to the conclusion that Spring was the most complete framework that would allow us to easily integrate the three separate parts into a final product. As such, we decided to use Spring Data JPA on the server because it provides repository support for the Java Persistence API (JPA), which simplified the development of Entities and integration with a database.

We chose to use a database instead of storing data in a text file because not only is it best practice and will give us a nice foundation for future professional experiences, but also a DBMS like PostgreSQL provides us with a more robust system with which to store and manage our data. Indeed, we chose to work with relational databases over NoSQL alternatives because of the ACID guarantees it provides us, and we decided on PostgreSQL in particular over other relational databases because we all had some previous experience with it (mainly gained via the Web and Database Technology course from the previous quarter).

Integrating a relational database with Spring for an application that communicates with the client via JSON objects proved more challenging than it would have been with MongoDB, given its document-based structure. However, a nice advantage was gained when deploying the server and database on Heroku, which had very nice support for PostgreSQL in particular. We decided to use Heroku in order to facilitate the process of accepting connections from and serving multiple users, and ensuring data consistency across the entire team.

Also on the server side, we decided to use the API provided by the CoolClimate Network for calculating the CO₂ emission that each one of our features would save because their numbers would be more accurate than if we decided to carry out our own research on the subject. Moreover, our project can automatically benefit from any future updates they choose to make to their numbers, further improving our accuracy. However, in the cases where the API could not provide the information we needed, we did do our own research to come up with formulae that could provide us with the necessary numbers, accurate to the best of our ability.

On the client side, we decided to use Spring's RestTemplate builder for establishing a connection and communicating with the server because not only did it facilitate this side of the work, but also enabled us to mock these connections in order to test them with Mockito. Indeed, we chose to use Mockito for testing both the client and the server because it guaranteed that no alterations would be made to the data in our database, and because implementing an embedded database for tests proved too complex a task for the timeframe we were given.

For the Graphical User Interface, we chose to use JavaFX over Swing because not only is Swing deprecated, but also event handling is not supported in Swing to the extent needed by our application. Moreover, JavaFX provided us with SceneBuilder, which allowed us to create a more aesthetically pleasing GUI with more ease.

Process

Due to our relative inexperience, more often than not, the large majority of the work to be done proved to be more complex than what we had anticipated, so we had to remain flexible in order to change and adapt accordingly. We also had issues planning ahead of each demo, as splitting the multiple tasks was often misjudged and more responsibility was assigned to certain members than with which they could cope. This, in turn, required others to offer assistance or take over the tasks entirely with very little time left before some demos in order to guarantee that all requirements were met. Nevertheless, despite these problems, we always managed to finish all the work needed for the demos.

From a somewhat objective perspective, it is clear that communication was an issue for the team as a whole. We encountered difficulties when communicating amongst members working on the same part of the project. Notwithstanding, we managed to overcome the issues caused by miscommunication ahead of each deadline. This is mainly due to the fact that team members have been both willing to help each other when help was needed and understanding of others' failures.

Most of our communication with each other was done via a WhatsApp group we created at the very beginning of the project, whilst communication with our TA was carried out mainly in person during the assigned Monday group meetings and sometimes via *Mattermost*.

Version control - namely, Git - was very helpful in allowing us to work on separate branches simultaneously, so that different parts of the application could not only be developed at the same time but also (whenever possible) did not depend on someone else finishing their task before it could be started. As a byproduct of its own *raison d'être*, Git also allowed us to revert commits and merges whenever we accidentally pushed directly to master or included code that was unfinished.

The main lessons learned from this project are related to gaining experience in working with a small-scale team project from beginning to end: from the good practices involved in team-based projects to the challenges involved in integrating the separate pieces of such a project. Therefore, it has been a rather effective way of preparing us for larger, future projects and, eventually, the workplace, where the complexities involved in each aspect are bound to be magnified, from research through implementation to release.

Reflection

Regarding the product, a possible point of improvement would be to provide more consistency through using centralised classes on the server side. For instance, a controller abstract class could be created and used as a template from which specialised instances could be derived. However, this would be a rather cumbersome task, as controllers differ considerably from one another. Thus, we decided it would only be a worthwhile endeavour if the project were to be scaled significantly. Moreover, the GUI could indubitably be improved with regards to its visual and functional design, yet, due to the reality of the project and our limited experiences and timeframe, we decided to keep things simple and execute them to the best of our abilities. Lastly, carrying out integration tests would unquestionably diminish the probability of our application having bugs and behaving in unexpected ways because it would ensure that it works seamlessly from end to end. Nevertheless, the complexity of setting up such testing environments proved to be beyond the realm of feasibility for this project.

Collaboration could be improved by agreeing on common coding sessions during a sprint in which all member meet to make sure that everyone has a clear, uniform idea of the current state of the project and which aspects require more attention and more work, and subsequently plan their own work accordingly. In order to facilitate this, the mandatory weekly meeting could be divided into two (*i.e.* two meetings of two hours each instead of one four-hour meeting). Another possible way of tackling collaboration would be to divide the work differently: whereas we chose to divide it by functionality (*i.e.* server side, client side, and GUI), we could instead divide it by feature, so that each person works on a single feature from end to end. This could help prevent that two people work on the very same task simultaneously.

Although we understand that the project was designed to simulate a real professional environment, nevertheless we feel that clearer guidelines and specifications, more consistency in the information present in different sources, and more explicit and draconian feedback from supervisors would greatly improve it. For example, more specific recommendations and requirements regarding the necessary code contribution or the distribution of labour would have been greatly appreciated. Moreover, whilst the project provided considerable freedom in general, the predetermined features to be implemented were somewhat misguided in their compatibility with the requirements - e.g. it proved rather challenging to calculate the amount of CO₂ emissions saved from lowering the temperature of your house just once. Lastly, some tuition on more technical topics would have been appreciated.

Individual Feedback

Calin Georgescu

The experience I have gained through this project has been varied and impactful with regard to both the way I approach team-based projects and the way I communicate with the members of my team.

Some of my weak points have indeed negatively influenced the team. My inability to coordinate with the other team members has led to an inefficient workflow and a terrible distribution of labour which required unnecessarily much time to fix, which could have been fixed from the get-go if we had communicated more efficiently.

As far my strong points are concerned, in the development plan I listed structured workflow and communication skills. While the organised workflow has been useful in managing the tasks I was assigned, I overestimated the ability to efficiently communicate with the rest of the members. As a result, there was a lot of work which needed to be done far later than we had initially intended it to be, which constitutes a significant part of the aforementioned lack of coordination.

Conflicts between team members have not been an issue as although we each had individual issues we either managed to come together and fix them or to work them out individually, and we both agreed on the broad direction our project was going, therefore eliminating the issue of conflicting views.

In conclusion, I believe this project has been an excellent opportunity to learn the basics of working in a team, along with its advantages, challenges and rewards, as well as providing a realistic basis from which to form basic expectations about the future of our careers in the domain.

Danila Romanov

I first created a database that was going to be used before it was converted and its schema is that which was used in the server. After that I switched roles and started working on the GUI and client. My notable contributions there is creating the friends page and implementing methods that would allow the communication between server and client. I have also implemented some other parts like registering a new user, and worked on improving the different parts of the GUI.

I would say my strong points in the project was being active in the group discussions as well as being adaptable with what I can work on. Some problems that I had came from communication, where I would be working on something and finished it and found that someone had done it too (which happened with the menubar and transition). Another problem was after I finished working on

the database, it was a bit difficult trying to find a place where I could work on something, as the main focus was mainly on the server and it already had a lot of people working on it. Outside from that, I find it was not difficult to find something to work on and give my contributions.

Dorka Hévizi

My strong points during the project were my ability to compromise and my persistence.

I did not have many conflicts with my teammates, it is partly due to the fact that I worked mostly on the API by myself and calculating CO₂ emissions, and my team members were all very pleasant people to work with. I am dissatisfied with the amount of my contributions, and despite feeling like I have worked hard I know that my work does not show it. I have feared getting in the way of my fellow teammates and my feelings of inadequacy led me to feel like I could not be useful. I have also been intimidated by the thought that my shortcomings would also hinder others, so I was afraid to ask for or offer help.

This project helped me see how something is built from nothing and I find complex problems a little bit less intimidating after this. I felt lost and overwhelmed for most of the quarter however.

The topic being something I am highly interested in and enjoy researching, I found it frustrating to not be able to meet my own expectations of the application.

After the project I still have a lot of issues with my own work ethic and in general I am underwhelmed with my own achievements and progress. I have looked into ways to start working on my shortcomings, but I cannot see myself overcoming them overnight without help.

Francine Biazin do Nascimento

This project in its entirety has been a very unique experience, which I had not quite encountered before. Even though I have worked in Project Management as a Junior Analyst, I do not feel as though the two can be directly compared. It has been rather informative and quite enlightening not only with regard to what it is like to develop an application from beginning to end, but also about how I work within a team.

I have learned that some of my core strengths sometimes deteriorate into my main weaknesses, such as allowing my perfectionism to refrain me from allowing other people to take over certain tasks because of my own apprehension that these tasks would not be performed correctly. My own punctuality and sense of responsibility also drove me to keep track of all deadlines and constantly remind people of what was due when in order to ensure that nothing would be overlooked and no deadlines would be missed. Even so, I do believe that I managed to restrain myself so as to neither micromanage anyone nor impose my own ideas of what the project should be like on the whole team.

Furthermore, overall, I have used my strengths and interests to our collective advantage: given my disciplined disposition and desire to learn more about Scrum, I volunteered to be Scrum Master and ensure that we always had a representative Scrum Board, and that all the necessary reports and documentation were uploaded to their respective folders on GitLab.

Personally, this project has been very instructive in highlighting some similarities, but primarily the many differences between working in a project as a Software Developer in an academic environment and as a Junior Analyst in Project Management in a professional environment.

Pradhyumnaa Ganapathi Subramanian

I exclusively worked on the Graphical User Interface of the project as it was an area where I felt confident and additionally, had a short term experience with while I used a different language in the past.

Approximately three days before each week's demo, I used to code and push the GUI needed for the demo. This way, the programmers from the client-side would have sufficient time to connect and test the interaction of the client code after connecting it with the respective buttons of the User Interface.

Before I start mentioning my strengths, I shall discuss about my weaknesses. During the week of the third demo, a crucial miscommunication from my part led to my contributions for the third demo to be rather limited. This is something that I wish to rectify.

Up until demo three, I was quite punctual with my work and submitted all the code that was required for the demo on time. I also came up with a few "absurd" ideas such as the login screen playing music in the background and a "welcome video" to get the users know about what the application does and how to use it. (These were not only ideas, I actually implemented both of them).

I did encounter a few problems during the development of the User Interface. The first week, I was debating on whether I should use Swing or JavaFX. I ended up choosing Swing first but soon realised that its capabilities are rather limited. Then, I had to switch over to JavaFX. When I did, I did not know that a program called SceneBuilder existed so I had to manually code in the positions of the buttons, the text fields and also the password fields. When I found out that such a program existed, I had to scrape all the progress of my first two weeks and then start from scratch so I could make the GUI ready by the time of the second demo.

Yash Kalia

It has been a tremendous learning curve and a true learning experience working on the project. Given the pacing and the sprint method I was able learn how to function efficiently individually in a team. Furthermore I learned a lot about applications like JavaFX and client server communication.

Also I increased my knowledge of databases.

Seeing everyone on the team working hard on the project inspired me to do the same and the level of cooperation and help I got from my team members was remarkable.

I also learnt a lot about essential applications like Git which was a personal goal for me.

I worked on the GUI for the application thus helping shape what the application really looked like and also was part of the client - therefore in charge of making sure that the client and server worked seamlessly and that our application had as many features as possible to make it more versatile and provide a great experience to the users.

Additional features from my side included creating a personal progress page for the users where they could view their progress in saving on CO₂ emission along with sorting the users according to their ranking among their friends in the leaderboard option.

All in all it has been a truly educative experience working on the project with my fellow teammates and I am very grateful that they helped me with my problems while working on this application.

Value Sensitive Design

The main values of our application are to not only environmental (*i.e.* helping people reduce their CO₂ emissions), but also financial (*i.e.* helping them reduce their spending). As such, our users would be saving money when not buying petrol for the car, reducing electricity and heating bills, sparing the cost of meat and expensive imported produce, amongst other examples. And while our main goal is to make people more conscious about the environment and how much their simple everyday choices impact the whole planet, it also encourages people to act outside of their comfort zone with a friendly, yet slightly competitive medium: a game. The gamification helps make people feel rewarded for their actions, which would usually only, if ever, show results in a month or year's time.

One other value which we would like to design for were we given the opportunity to redo this project from scratch in a real, professional environment, would be data transparency. As such, we would strive to ensure that every user is aware of what kind of data we are gathering from them

whenever they use our application and how we are storing it, using it, and keeping it secure. We would not request any superfluous data, such as gender, sexual orientation and address, so that we can avoid any sort of unintended discrimination and minimise the harm caused by any potential data leaks caused by hacker attacks. We would allow each user to download their own data from our database so that they have complete access to everything we have stored concerning them.

In order to ensure that this process is done properly and legally, we would employ lawyers to get us up to code with GDPR regulations and advise us on best practices. We would also consult psychologists who could give input on how to approach this subject with users so as not to inadvertently imply that we are data mining - which is the very opposite of what we intend to do.

One possible tension which might result from designing for these values is that, in order to maximise our reach and potentially create a real impact which might help the environment and also benefit the lives of our users in leading a more sustainable lifestyle, we would have to secure some form of venture capital to fund the project. For technology-related products, this is usually found in the form of data mining, or by placing advertisements in our application, which would almost certainly employ third-party cookies which could be used to track users across different online platforms. As such, this would be in tension with our efforts to provide data transparency.

One possible mitigation for this tension would be to ensure that the funding we secure for our project does not require any form of data from our users, and that no advertisements are displayed in our application.