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

The first major decision made 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, it was concluded that Spring was the most complete framework that would allow for the integration of the three separate parts into a final product. As such, it was 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.

Use of a database instead of storing data in a text file was chosen because not only is it best practice and will give the team a nice foundation for future professional experiences, but also a DBMS like PostgreSQL provides the application with a more robust system with which to store and manage data. Indeed, it was chosen to work with relational databases over NoSQL alternatives because of the ACID guarantees it provides, and PostgreSQL in particular was chosen over other relational databases because the entire team 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 native support for PostgreSQL in particular. It was 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, the decision to use the API provided by the CoolClimate Network for calculating the CO_2 emission that each one of the features would save was made because CoolClimate's numbers would be more accurate than if the team had decided to carry out its own research on the subject. Moreover, the application can automatically benefit from any future updates CoolClimate chooses to make to their numbers, further improving the application's accuracy. However, in the cases where the API could not provide the information needed, research was undertaken to come up with formulae that could provide the necessary numbers, accurate to the best of the team's ability.

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

For the Graphical User Interface, JavaFX was chosen over Swing because not only is Swing deprecated, but also event handling is not supported in Swing to the extent needed by the application. Moreover, JavaFX provided the team with SceneBuilder, which allowed for the creation of a more aesthetically pleasing GUI with more ease.

Process

Due to the team's relative inexperience, more often than not, the large majority of the work to be done proved to be more complex than what had been anticipated, so the team had to remain flexible in order to change and adapt accordingly. There were also issues with 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, the team 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. There were difficulties when communicating amongst members working on the same part of the project. Notwithstanding, the issues caused by miscommunication were overcome 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 as well as understanding of others' failures.

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

Version control - namely, Git - was very helpful in allowing the team 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 one to revert commits and merges whenever one accidentally pushed directly to master or included code that was unfinished. It also provided the team with the opportunity to learn about and practice code reviews, which are standard in a professional environment. This has also helped to ensure not only that everyone involved has checked and understood a piece of code completely, but also that potential bugs are caught as early as possible and rectified before the code is merged to master.[1]

[1] The team's code reviews are best exemplified by the following merge requests: !55, !57, !63, !80, !83.

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 associated with 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 the team 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, it was decided that 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 the team's limited experiences and timeframe, it was decided to keep things simple and execute them to the best of each member's abilities. Lastly, carrying out integration tests would unquestionably diminish the probability of the application having bugs and behaving in unexpected ways because it would ensure that it works smoothly and continuously 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 it was chosen to divide it by functionality (*i.e.,* server side, client side, and GUI), it could have been divide instead 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 it is understood that the project was designed to simulate a real, professional environment, nevertheless it is felt that clearer guidelines and specifications, more consistency in the information present in different sources, and more detailed 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 CO_2 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 Calin has gained through this project has been varied and impactful with regard to both the way he approaches team-based projects and the way he communicates with the members of his team.

Some of his weak points have indeed negatively influenced the team. His 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 the team had communicated more efficiently.

As far as his strong points are concerned, in the development plan he listed structured workflow and communication skills. While the organised workflow has been useful in managing the tasks he was assigned, he 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 it had initially been intended to be done, which constitutes a significant part of the aforementioned lack of coordination.

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

In conclusion, Calin believes 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 future careers in the domain.

Danila Romanov

Danila 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 he switched roles and started working on the GUI and client. His notable contributions there are creating the friends page and implementing methods that would allow the communication between server and client. He has also implemented some other parts like registering a new user, and worked on improving the different parts of the GUI.

He would say his strong points in the project were being active in the group discussions as well as being adaptable with what he can work on. Some problems that he had came from communication, where he 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 he finished working on the database, it was a bit difficult trying to find a place where he could work on something, as the main focus was mainly on the server and it already had a lot of people working on it. Other than that, he finds it was not difficult to find something to work on and give his contributions.

Dorka Hévizi

Dorka's strong points during the project were her ability to compromise and her persistence.

She did not have many conflicts with her teammates, and this is partly due to the fact that she worked mostly on the API by herself and calculating CO_2 emissions, and her teammates were all very pleasant people to work with. She is dissatisfied with the amount of her contributions, and despite feeling like she has worked hard she knows that her work does not show it. She has feared getting in the way of her fellow teammates and her feelings of inadequacy have led her to feel like she could not be useful. She has also been intimidated by the thought that her shortcomings would also hinder others, so she was afraid to ask for or offer help.

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

The topic being something she is highly interested in and enjoys researching, she found it frustrating to not be able to meet her own expectations of the application.

After the project she still has a lot of issues with her own work ethic and in general she is underwhelmed with her own achievements and progress. She has looked into ways to start working on her shortcomings, but she cannot see herself overcoming them overnight without help.

Francine Biazin do Nascimento

This project in its entirety has been a very unique experience for Francine, which she had not quite encountered before. Even though she has worked in Project Management as a Junior Analyst, she does 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 she works within a team.

She has learned that some of her core strengths sometimes deteriorate into her main weaknesses, such as allowing her perfectionism to refrain her from allowing other people to take over

certain tasks because of her own apprehension that these tasks would not be performed correctly. Her punctuality and sense of responsibility also drove her 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, she does believe that she has managed to restrain herself so as to neither micromanage anyone nor impose her own ideas on what the project should be like upon the whole team.

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

For Francine, 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

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

Approximately three days before each week's demo, he 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 he starts mentioning his strengths, he shall discuss his weaknesses. During the week of the third demo, a crucial miscommunication from his part led to his contributions for the third demo to be rather limited. This is something that he wishes to rectify.

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

He did encounter a few problems during the development of the Graphical User Interface. During the first week, he was debating on whether he should use Swing or JavaFX. He ended up choosing Swing first but soon realised that its capabilities are rather limited. Then, he had to switch over to JavaFX. When he did, he did not know that a program called SceneBuilder existed so he had to manually code in the positions of

the buttons, the text fields and also the password fields. When he found out that such a program existed, he had to scrape all the progress of his first two weeks and then start from scratch so that he could make the GUI ready by the time of the second demo.

Yash Kalia

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

Also he increased his knowledge of databases.

Seeing everyone on the team working hard on the project inspired him to do the same and the level of cooperation and help he got from his teammates was remarkable.

He also learnt a lot about essential applications like Git which was a personal goal for him.

He 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 the application had as many features as possible to make it more versatile and provide a great experience to the users.

Additional features from his side included creating a personal progress page for the users where they could view their progress in saving on CO_2 emission along with sorting the users according to their ranking among their friends in the leaderboard option.

All in all fir Yash it has been a truly educative experience working on the project with his fellow teammates and he is very grateful that they helped him with his problems while working on the application.

Value Sensitive Design

The main values of the application are not only environmental (*i.e.,* helping people reduce their CO_2 emissions), but also financial (*i.e.,* helping them reduce their spending). As such, the users would be saving money when not buying petrol for the car, reducing electricity and heating bills, and sparing the cost of meat and expensive imported produce, amongst other examples. And while the main goal was 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 through a friendly, yet slightly competitive medium: a game. The gamification helps to make people

feel rewarded for their actions, which would usually only, if ever, show results in a month or year's time.[2]

One other value which the team would have liked to have design for were it given the opportunity to redo this project from scratch in a real, professional environment, would be data transparency. As such, the team would strive to ensure that every user is aware of what kind of data is being gathered from them whenever they use the application and how it is being stored, used, and kept secure. No superfluous data would be requested, such as gender, sexual orientation and address, so that any sort of unintended discrimination would be avoided and any harm caused by any potential data leaks caused by hacker attacks would be minimised.[3] 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.[4] 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.[5] 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.

---

[2] Indeed, any real impact would require global cooperation and, perhaps, an economical "revolution," as Jonathan Park discusses in his paper, "Climate Change and Capitalism," *Consilience*, no. 14 (2015): 189-206.

[3] This is particularly important because users are generally not equipped with all the information necessary to make informed decisions about their privacy, as Alessandro Acquisti, Curtis Taylor, and Liad Wagman argue in "The Economics of Privacy," *Journal of Economic Literature* 54, no. 2 (2016): 442-92.

[4] For a discussion on the tension between right to privacy and right to data protection, see Shraddha Kulhari, "Data Protection, Privacy and Identity: A Complex Triad," in *Building-Blocks of a Data Protection Revolution: The Uneasy Case for Blockchain Technology to Secure Privacy and Identity* (Baden-Baden, Germany: Nomos Verlagsgesellschaft MbH, 2018), 23-37.

[5] For a brief discussion on how advertisers target users in technological platforms for profit, see Annika Richterich, "Big Data: Ethical Debates" in *The Big Data Agenda: Data Ethics and Critical Data Studies* (London: University of Westminster Press, 2018), 33-52.

Bibliography

Acquisti, Alessandro, Curtis Taylor, and Liad Wagman. "The Economics of Privacy." *Journal of Economic Literature* 54, no. 2 (2016): 442-92.

Kulhari, Shraddha. "Data Protection, Privacy and Identity: A Complex Triad." In *Building-Blocks of a Data Protection Revolution: The Uneasy Case for Blockchain Technology to Secure Privacy and Identity*, 23-37. Baden-Baden, Germany: Nomos Verlagsgesellschaft MbH, 2018.

Park, Jonathan T. "Climate Change and Capitalism." *Consilience*, no. 14 (2015): 189-206.

Richterich, Annika. "Big Data: Ethical Debates." In *The Big Data Agenda: Data Ethics and Critical Data Studies*, 33-52. London: University of Westminster Press, 2018.