



Group 5:

Duroyon Marin 4872355

Grishkov Ilya 4770811

Hilea Razvan 4941837

Lanzini Edoardo 4925122

Table of Contents

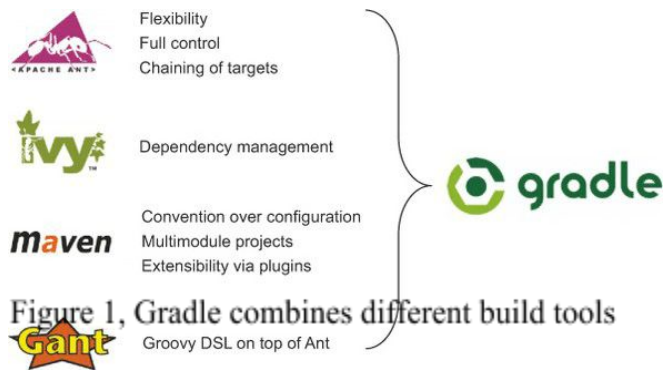
<i>Product:</i>	2
General	2
Client:	2
Server:	3
<i>Process:</i>	3
Weekly meetings:	3
SCRUM:	4
Communication:	4
<i>Reflection:</i>	4
Process:	4
Product:	4
Course:	4
<i>Individual feedback:</i>	5
Razvan:	5
Marin:	5
Ilya:	5
Edoardo:	6
<i>Value Sensitive Design:</i>	6
<i>Bibliography</i>	7

Product:

General:

This section describes the different design choices that affected both Client and Server.

The framework used in this project is Spring. Compared to other frameworks, Spring offers a wide variety of features useful to tackle the different parts of this project. Two important components that resulted to be particularly useful are Spring RESTful properties and Spring JPA. Through the use of Spring JPA and the easy-to-use object-relation mapping, the database and application collaborated seamlessly. After an easy integration setup, the app could benefit from the powerful database transaction management capabilities. Moreover, Spring allows to handle HTTP requests with the user of a controller. All in all, Spring is a resourceful framework that allowed us to integrate with other libraries while offering a RESTful web service.



The second design that affected both Client and Server was the use of **Gradle** as the default building tool for our project. Gradle is a Java-based build tool that allows for declaratively modeling your problem domain using a powerful and expressive domain-specific language (DSL) implemented in Groovy instead of XML (Muschko, 2015). As the picture on the left shows, Gradle combines the best features from other builds tools.

Client:

This section describes the different design choices related to the Client.

FXML: FXML is an XML-based language that provides the structure for building a user interface separate from the application logic of your code (Fedortsova, 2014). Thanks to the transparent scene graph, it is easy to create and manipulate user interfaces across the team. Moreover, the JavaFX Scene Builder makes it easy to tweak the GUI and offer the user the best experience (figure 2).



Server:

This section describes the different design choices related to the Server.



Figure 3, PostgreSQL + AWS RDS

Database: The database supporting envy.me is managed with PostgreSQL, a powerful, open source object-relational database system that uses and extends the SQL language. PostgreSQL offers many features to support the application development process, allowing env-y.me to protect and build its internal database. The deployment of the database was completed through Amazon Database Service (ADS). ADS gives the possibility to run PostgreSQL database engine (figure 3) offering a range of benefits, including **availability, scalability and performance** (Blaisdell, 2015)

Deployment: In order to deploy envy.me, Heroku and Amazon RDS were used. The server behind the app is running on Heroku and can be found at <http://env-y-me.herokuapp.com/>. Thanks to the cloud platform, the client is able to make HTTP requests without running a local server. Heroku is a great option to start an app right away, without much customization of the architecture (Rao, 2012). Thanks to its built-in tools and pre-built architecture, the setup was effortless.

Last, **Mockito** was used in order to test the integration of the different components and to **mock** the different parts connected in the project. Mockito was voted as the best mocking framework for Java (Mockito, 2019) thanks to its easy to use testing tools.

Process:

Weekly meetings:

The team was meeting twice a week, on Monday and on Thursday from the 11th of February until the 8th of April. During the first weekly meeting, the team discussed with the TA possible improvements and impediments, while also giving **demos** of the app at determined deadlines. At the same time, most of the meeting on Monday was spent during **Sprint Review** followed by a **Sprint Planning**. First, each team member would share the following:

- 1. What did you work on this week?
- 2. Were there any impediments in your way?

After that the team proceeded with the planning phase, splitting up the jobs into actionable tasks and then assign them to each team player based on their area of focus (see SCRUM Section).

Once all the tasks were assigned, the team engaged in a **Sprint Retrospective**. At this stage, the focus was on possible improvements, tackling those questions:

- 1. Are there any improvements to be made?
- 2. How can we do work more efficiently?

During the second weekly meeting, usually taking place on Thursday, the team discussed how the development of the features was proceeding, solving merge requests and connecting different parts of software. The intermediate weekly meeting proved to enhance the quality of code and productivity of the team, having the deal with less misunderstanding at a later stage.

SCRUM:

The use of a SCRUM board (figure 4) played an important role during the planning process. None of the team members had experience before with using this planning tool. Therefore, we decided it was better to assign the role of SCRUM Master to the chairman of the meeting so that everybody could keep a general view of the project structure and progress.

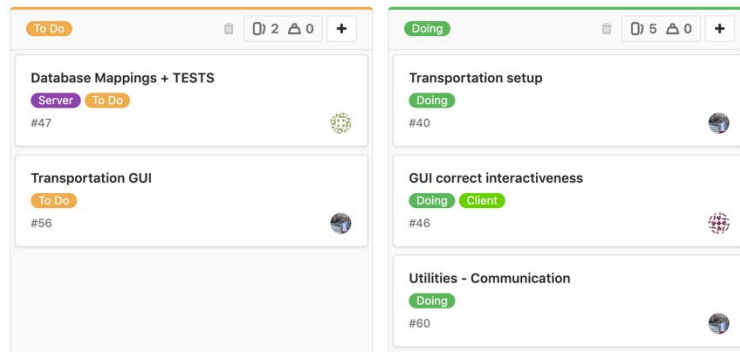


Figure 4, Snapshot of SCRUM Board

the progress in each component of the project.

For each task, the team gave an estimate of the number of hours it would take to tackle it. Thus, everyone would have the same workload estimate per week (~12 hours/week). At first, the error on the estimation was large but it decreased when nearing the end of the project. The team used GitLab Issues' Board as a SCRUM Board updating it as the development went along. Every opened issue was closed linking it to the respective Merge Request that included the code tackling the task. All in all, the SCRUM methodology proved to be effective while giving a general view of

Communication:

The team used two main mediums for communication. The first one is WhatsApp and the second one is Mattermost. 95% of the team communication went through WhatsApp while the remaining 5%, mostly concerning the communication with the TA, was completed using Mattermost. WhatsApp appeared to be an easy and effective solution for our needs. The team also discussed the use of other potential messaging platforms, such as Discord or Slack but decided to use WhatsApp given the initial effectiveness.

Reflection:

Process:

At times, communication within the team was lacking. Everyone was focusing on their own part without necessarily sharing possible design decisions that would affect the rest of the project. During the second and third week, this problem delayed the team schedule. Since then, the intermediate meeting allowed the team to solve the issue by increasing communication.

Product:

First, from a security point of view, the backend could definitely be improved. For instance, if a malicious user gets into possession of the source code, he/she could send POST and GET request to specific URLs and easily alter information in the Database.

Secondly, the goal of the team was to make it as easy as possible for the user to store information and “**play the game**”. Therefore, we discussed possible integrations with third-party apps such as Google Maps and My Fitness Pal that would offer the user a better experience. By integrating Env-Y ME! with other applications, the server could retrieve live information without the need for the user to manually fill in boring forms. In general, the team long-term vision was to create a platform where the user could see how his/her everyday activities affect his/her CO₂ consumption while at the same time competing with his/her friends.

Course:

The course could be improved by offering the students more information regarding how to use build tools such as Maven and Gradle. The team struggled initially with the user of Maven.

Moreover, it would be useful to have a set of “Getting Started” resources for the students, with tutorials taken from the internet.

Individual feedback:

Razvan:

I started this project by working on the database, but after a while, I started working solely on the client side. After working on the first version of the GUI, I shifted towards implementing functionality, which required a basic understanding of the inner workings of the server.

At first, it seemed like the project was going to be a complete failure, as we lost three team members in the first two weeks. However, this seemed to motivate the remaining team, and we actually managed to create a quality product. It also helped communication-wise, as the meetings were usually quick and on point.

One of my strong points that proved helpful during the project was my obsession with order. During the meetings, I was usually writing down all suggestions brought up, so we could discuss them all and not miss one. The weak point I managed to work most on during this project was my procrastination tendency. While I still put things off, I always managed to get the work done and not miss any deadlines.

Although I didn't get to work so much towards my goal of understanding servers, I did take part in a project where my teammates were invested in their work, which gave me a taste of how a big project should develop.

Marin:

Throughout the development of this project, I had a polyvalent role, contributing to different parts of the application. I started out working on the development of the server and building an initial basic graphical user interface. Soon enough I was associating the communication between the server and client, in order to enable a reliable connection.

Overall, we had a cohesive group with hard workers. I believe all team members could rely on everyone else, this allowed the project to grow steadily and efficiently. Nevertheless, the group still had some problems to inform the consistency of the code on either client or server side. Consequently, leading to some inconvenience while combining both sides. Apart from this issue, the team managed to overcome the lack of members and develop a stable ecological application.

My learning goal for this project was to improve collaboration and thinking before programming. To a certain extent, I believe I fulfilled these goals, spending more time on the design of the application before building and trying to fairly separate the different tasks. Moreover, with this process of building an application, I realized that I often need more time than others to plan out and program. I handled this by reserving a couple more hours to organize my work. In the end, this experience improved my skills as a programmer and initiated me to the fascinating world of software development.

Ilya:

During the project, my primary emphasis was shifted towards the development of the client-side application and the corresponding research needed to accomplish this task. The main objective was to make self-explanatory, user-friendly interface and well-working functionality behind it, that had the least possible number of unexpected behavior occurrences. While working on the client application, I still had to understand the functionality of the server to create the most efficient and reliable code.

All members of the group had passion and will to work on the project. While being the group of 4 increases workload, it also improves communication between its members, which was beneficial during the part, when multiple modules are being merged. Still being an issue, merging became less complicated, when the entire team had a strong sense of each other's tasks and their compatibility. Such a result would have been more difficult to

achieve in a larger group. Other than that, the development process went smoothly, and the workload was well distributed.

My final goal was to get more experience working with people, whom I didn't know before. I believe the project gave me the opportunity to work on the weak points I had, fulfill the goal, and further develop a sense of responsibility in front of the team for following deadlines and contributing to the project.

Edoardo:

Throughout the project, I focused on the back-end of the product, working on the server, the Database and the integration between the two.

My stronger points included communication and positive attitude. These skills definitely came into play during the project. During the meetings, the communication was efficient, and the team was motivated throughout the whole execution.

My weaker points included lack of experience and lack of details. The former translated into a steep learning curve, while the latter became an obsession with testing. I had no previous experience working on big projects and this allowed me to learn and absorb a lot in the past three months. I understood how the different components of the project fit together and how to break down big projects into smaller tasks.

Overall, the team showed great synergy and work ethic. All deadlines were met, and the communication was good. What I liked the most about the team was the attitude towards making something useful. Since the first meeting, our goal was not to use the guidelines as a checklist but as an inspiration and starting point to create something that everybody would want to use. I believe the final product shows this mentality. Throughout the project, everybody showed great initiative and the ability to take responsibilities.

Value Sensitive Design:

It is already well established that us humans have irreversibly altered the environment and that we need to change our lifestyle to reduce our negative impact on our planet.



Figure 5, Value Sensitive Design

Env-Y ME! aims to encourage people to change their lifestyle by allowing them to track their CO₂ consumption. Users can add friends and compare their carbon footprint, while also receiving achievements and badges for certain milestones, such as eating 250 vegan meals or travelling 1000 km on foot or by bike. The application stores each user's data on a secure database, which can only be accessed through our server, so each user's information is safely stored. The password of each user is encrypted, to mitigate security threats. Currently, the user cannot delete his/her information stored in the system unless cleared by the developer team. Thus, privacy concerns may arise in the case where the user wants to delete his account.

The development of this application was initiated by the realization that many people are aware of the detrimental ecological impact humans have on this planet, but they lack the drive to change their lifestyle. Env-Y ME! wishes to trigger this initiative by providing a social network where users compete with their friends to have the best score. The stakeholders of this application are the people that desire to switch to a ecological friendly lifestyle but achieve more with the encouragement of their friends. Furthermore, Env-Y ME! is also designed for users that want to track their CO₂ emissions, since the application can track many different types of

CO2 discharges. Users can choose to record from six different types of meals with up to eight different ingredient types, four different means of transportation and three house utilities.

The reward system (achievements, badges, and score) may encourage some users to exaggerate changes in their lifestyle. As a consequence, sudden changes, whether in dieting or transportations, may be harmful for the users. This problem, for example, could be solved by a “tip of the day” system, where our application could automatically analyze the user’s progress and help them switch to a new healthy and ecological friendly life without taking drastic measures. Therefore, it could avoid potential damaging consequences of overdoing and overusing the application.

Last, the scoring relies on a peer anti-cheat system, where every user can see what their friends did to achieve a certain score. This factor should keep users away from merely lying when adding data to their profile. Moreover, this system allows to build a community that will support each other as peer reviewing will encourage the users to limit their CO2 emissions by comparing themselves to their friends.

There are currently no options that are meant to help people with certain disabilities, such as visual or mental. However, we believe that such features are very important, especially since our application is designed to be a gamified social network of people that aim to change themselves in order to save our planet. Therefore, we plan on adding special information tooltips on each of our pages, to help users navigate easier through the application, and also special color modes for color blind people.

Bibliography

- Blaisdell, R. (2015, March 24). *Benefits for using AWS RDS*. Retrieved from Rick's Cloud:
<https://rickscloud.com/benefits-for-using-aws-rds/>
- Fedortsova, I. (2014, March 17). *Why use FXML*. Retrieved from Oracle:
https://docs.oracle.com/javafx/2/fxml_get_started/why_use_fxml.htm
- Mockito. (2019, January 24). *Tasty mocking framework for unit tests in Java*. Retrieved from Mockito:
<https://site.mockito.org/>
- Muschko, B. (2015, July 8). *Why Build Your Java Projects with Gradle Rather than Ant or Maven?* Retrieved from Dr. Dobb's: <http://www.drdobbs.com/jvm/why-build-your-java-projects-with-gradle/240168608>
- Rao, B. (2012, February 16). *Why do people use Heroku when AWS is present? What distinguishes Heroku from AWS?* Retrieved from Stack Overflow:
<https://stackoverflow.com/questions/9802259/why-do-people-use-heroku-when-aws-is-present-what-distiguishes-heroku-from-aws>