

# #GoGreen Report

Group 83



Lucas van de Geer – 4983521

Sami Farahi - 4723422

Shaan Hossain - 4908058

Alex Shulzycki - 4911148

Vanessa Timmer – 4728033

# Table of Contents

Table of Contents .....	2
Section 1: Product .....	3
1.1 - Technological Choices .....	3
1.1.1 - PostgreSQL.....	3
1.1.2 – JSON .....	3
1.1.3 – GSON .....	3
1.1.4 – Calculator API .....	3
1.1.5 – JavaFX.....	3
Section 2: Process .....	4
Section 3: Reflection .....	5
3.1 - Reflection on process .....	5
3.2 - Reflection on product.....	5
3.3 - Reflection on course .....	5
Section 4: Individual feedback .....	6
4.1 – Lucas van der Geer .....	6
Contribution.....	6
Weak points .....	6
Strong points.....	6
4.2 – Sami Farahi .....	6
Contribution.....	6
Weak points .....	7
Strong points.....	7
4.3 - Shaan Hossain .....	7
Contribution.....	7
Weak points .....	7
Strong points.....	7
4.4 - Alex Shulzycki .....	8
Contribution.....	8
Weak points .....	8
Strong points.....	8
4.5 - Vanessa Timmer .....	8
Contribution.....	8
Weak points .....	8
Strong points.....	8
Section 5: Value Sensitive Design .....	9
Design for Environmentalists .....	9

## Section 1: Product

### 1.1 - Technological Choices

#### 1.1.1 - PostgreSQL

We have chosen to use PostgreSQL. Our main reason was that we all are very familiar with PostgreSQL because it was widely used and discussed in the course Web and Database Technology from Q2. A huge benefit was that the GUI gives a clear overview of the database. Secondly, the major popularity of PostgreSQL brings many advantages such as easy and quick implementation with several cloud platforms. The connection between PostgreSQL and Heroku was very easy and quick. An alternative was MySQL, but it had two disadvantages as oppose to PostgreSQL. Firstly, only Shaan had experience with MySQL before, this would have cost most of our team a lot of ambiguous self-study trying to figure out how MySQL exactly works. Secondly, MySQL is not as easily implementable with Heroku as PostgreSQL.

#### 1.1.2 – JSON

JSON was used because it is the accepted standard for REST APIs. It is easy to use and widely supported. There was really no competition.

#### 1.1.3 – GSON

Google's GSON was used to both parse some of the JSON on the client side, and to parse JSON requests into their own Java classes based on a template. This allowed a much quicker and robust workflow, as each kind of request was processed in its own java class, and all the heavy lifting of initializing the class and giving it its respective JSON parameters was done by GSON. Once the framework was down, adding more functionality was extremely simple.

#### 1.1.4 – Calculator API

We chose to work with the API from BrighterPlanet, because they had all the aspects we needed for our project. We used the API especially for the Car vs Bike/Public transport features. Second for the vegetarian meal we decided to use fixed amount of points. Furthermore, for the 'Temperature adjustment' we searched about how much CO<sub>2</sub> you could save when changing the temperature of your house, from a high temperature to a lower temperature. And so, we used the API again for calculating how much CO<sub>2</sub> you would save per month when adjusting the temperature of your house. About 'Local produce' we found out that whenever you buy imported food (1 kilograms), in this case fruit and vegetarian food, you would emit 560 grams of CO<sub>2</sub>. However, when you buy local food (1 kilograms), in this case fruit and vegetarian food, you would emit 256 grams of CO<sub>2</sub>. So, we made a calculation where we would subtract these amounts from each other. As last about the 'Solar panels' we found that whenever you use solar panels, you would save 0.46 kilograms of CO<sub>2</sub> per 1 kWh (Kilowatt-hour).

#### 1.1.5 – JavaFX

For the Graphical User Interface, we chose to make use of JavaFX. Unlike other GUI frameworks, JavaFX can be used with Scenebuilder, which gives instant visual feedback. Another advantage that puts it ahead of other frameworks is that JavaFX can be used in the development of mobile applications. Although this project only required that we make a desktop application, the mobile market continues growing every day, so if given the opportunity to work with a framework that's relevant to both desktop and mobile applications we'll take it as it's a skill that might turn out to be useful in the future. JavaFX also allows for more in-depth customization by integrating CSS, giving us more control over the result.

## Section 2: Process

- Planning
  - We started our project with some difficulties, because nobody knew each other, and we did not really know what we could expect from the project. After our first meeting, we did not really plan something to do for each member, because we did not know what every member could do in the beginning. But after our second meeting and we got a little bit more used to each other, and we started to plan out what every member should be doing every week. Every week we looked back at what every member of the group did, and if he/she did have some setbacks. In this way, we managed to keep our work up to date, so we would reach every deadline without any big problems. In the end we are happy about how we planned out every task. We finished our application in time, we are glad of how it works, and it looks very good.
- Team collaboration
  - To make sure that we agreed on a JSON protocol, we created API-list.md where we put all the JSON requests that were in use.
  - To make sure that everyone knew how to write server-side logic for individual requests, instructions for writing server queries .md was written.
  - To finalize how exactly everything would work, we created a google doc with the descriptions of each request and functionality, and which data each needed.
  - To make sure everyone knew how to interact with the database, an ER diagram was created.
- Communication
  - Most communication on specific things was done via the WhatsApp group
  - The scrum board and other GitLab based tools were used to make sure everyone knew the bigger picture and knew what had to be done.
- Version Control
  - Git was used as required, branching was also used effectively, with merging usually done in person so that any questions could be immediately answered.
- What went wrong, what did we learn from that
  - We had issues with JavaFX for the first few weeks, as it did not want to work with all our IDE's. We also had some issues with JaCoCo and generally with everything maven-related. We managed to work through it with by asking our TA's, using a lot of google, and unhealthy amounts of caffeine. It really helped hone our problem solving and troubleshooting skills.
  - None of us ever used a CI/CD pipeline, so we had to basically try from scratch. We had a lot of issues, as with maven, however all it took was again, lots of time and perseverance.

## Section 3: Reflection

### 3.1 - Reflection on process

We as a group are quite happy of how we managed to get the result by working hard together. We split up every task over all the group members, and everyone finished what he/she had to do on time. When someone was stuck on something, we tried to help that person as good as possible, because that is what we should do as a group. The communication online could have been a bit better, because not always did everyone respond to something when we asked for everyone's opinion. This was not a huge setback, but it should improve for the next time we would work together.

### 3.2 - Reflection on product

In the end we are happy about our application. We have all the features that needed to be done. The way the application looks like is also really good, it is very clear and easy to use. Maybe we could have added some bonus features like the android app. But we wanted to make sure that we first would have a good working application with the obligated parts of the project. In conclusion we are very proud of what we have managed to create as a group.

### 3.3 - Reflection on course

We have all been waiting for what the subject would be for this project of this this course. After we got to know that the subject was '#GoGreen' we did not know what we could expect at all. But everything became clearer after a few lectures and days. The lectures were clear, but we would have preferred a bit more explanation about Git, because most people did not even know how Git worked or what it was. Also, about Maven a bit more explanation would be necessary to have. The freedom about choosing which IDE you want to use, was pretty good. The idea of figuring mostly everything self out, was also a good choice, because in this way we could learn more about looking things up ourselves, and eventually understanding the things we had to look up better. Overall the course was good and fun at the same time, we must work together as a group to get the result we wanted, and we did.

## Section 4: Individual feedback

### 4.1 – Lucas van der Geer

#### Contribution

The start of this project wasn't easy. A lot of the time I wasn't entirely sure what to expect or do, so in the beginning I didn't get a lot done. I also had the misfortune to fall ill for an entire week, which meant I couldn't work much on the project then either. Later, though I finally was able to get a lot done and make up for my slow start. I was responsible for all the gamification aspects and I feel like I managed to implement those well. For the first gamification aspect, comparing your tracked CO2 to your friends, I had to write code to control the FXML page so that it would actually display your saved CO2 per category and that of your friends as well. To do that effectively I had to write a separate query to instantly fetch all the user's CO2 saved per category in one request to the database, which increases efficiency, rather than sending different requests for all the categories. At that time, we also hadn't implemented a friend system yet, so I also had to find a way to get the global average of points. I wrote another separate database query for that, which not only fetched the total amount of points per categories for all users, but the amount of users as well. From there on out it was easy to calculate the global average amount of points and to display it on the GUI. I was also responsible for testing these queries.

#### Weak points

My weaker points were that I sometimes didn't communicate well or on time, which resulted in some confusion, though it wasn't drastic, and I feel like overall I have improved on my communication skills. Another weak point of mine was that sometimes I feel like I couldn't keep up, simply because I didn't really understand the code. I don't really have a background in programming, which I feel hindered me a lot during this project. A lot of the things we've had to do in this project we haven't explicitly been taught and I feel like a background in programming would have helped me understand how to do those things better.

#### Strong points

One of my stronger points was my perseverance. At some point during this project I genuinely thought I would have to drop out of the computer science course because I didn't feel like I understood anything. But I worked through that by spending a lot of time trying to better understand the project, until I had it all figured out again and could continue working on it.

### 4.2 – Sami Farahi

#### Contribution

At the start of the project I had no clue of what I could expect from this project. Especially with the random groups, I was not so sure how I would work together with people I did not know at all. At our first meeting everyone was a little bit quiet which made sense, because no one knew each other. But after a while we started to talk. We actually had seven members in our group, but two of them never showed up, so we ended up having only five members. This was not really a problem for us, because we all did our part that had to be done. I figured out that I did not have to be afraid at all to let the others know my opinion about things, which was new for me. As I stated in my personal development plan. I did not really know what I could do in the beginning, because there weren't many tasks. After our first demo, I realized that this project is nice to do. I worked well with the other member of the group, and the communication also went well. I decided that I would do the part of the API calculator. The TA told me that this could be the hardest part of the project, but I wanted to challenge myself, and see if I could figure out how it all works. The API calculator cost me a lot of time, I struggled a bit, but I managed to get it done. I was proud and relieved that I got it done, because it was a very important part of the project, and one of the hardest. Overall, I think

that my progress throughout this project was quite good, I developed my programming skills and my lack of letting others know my opinion. We did a good job together as a group, and I think we ended up with a very good Java application.

#### Weak points

I stated in my personal development plan that I sometimes don't really speak up, and that I don't really share my opinion. I think that I managed to improve these weak points and worked on them during this project. I noticed that when I spoke up or shared my opinion that there was nothing to be afraid about, which made me realize that I should do it more often.

#### Strong points

Two of my strong points are that I can work well in a group and that I am always willing to help others. In the beginning, I did not really know what I could expect from the group, but I had no doubts that I would be able to work well together with the other group members. Whenever someone asked me if I could help them, I tried my best to help them. I also offered sometimes if someone needed any help, or if they had any question or something, they could just ask them.

### 4.3 - Shaan Hossain

#### Contribution

I've worked on many things. At the start of the project many things were unknown to me. I think I can speak behalf my whole team that no one heard of Checkstyle and JaCoCo before. Once I figured out how they work I wrote some instructions to my team, guiding them how to install several drivers. In terms of task distribution, I was assigned on setting up and maintaining the database. Since I was the one with the most experience with databases it was the most logical option that I would be the one in charge of the database. Further on in the project I have also contributed in the GUI part with Vanessa. After the back-end was (more or less) finished by Alex I was in charge of the implementation with the GUI. JSON requests needed to be sent and received with a push of a button and the corresponding queries needed to be executed and printed. Once that was finished (or stable at least) I teamed up with Sami to give the API some finishing touches. Afterwards I made sure that the GUI was working correctly with the API and printed the correct result. At last writing test for all database classes and optimizing the code was the only thing left to do.

#### Weak points

I definitely fell into the same pitfall as I always do. I guess that is just a characteristic of mine. Underestimating the workload, I am assigning myself to is just an issue I will never be able to resolve. But to be completely honest, (sometimes) I had no choice to take up the work on me because otherwise no one would do them (, but I guess the analytics speak for themselves).

#### Strong points

I think taking the leadership (especially at the start of the project) really started to get the grinds of this team going. I was not completely in charge of course, but I definitely was one of the few people who kept communicating with the rest to make sure that we are all on the same page. My drive to reach my goal has also really paid off for this project. A lot of research had to be done, especially during the first few weeks. I have made some long shifts to make my code working and as optimal as possible. But those late nights/early mornings were definitely worth it in the end.

#### 4.4 - Alex Shulzycki

##### Contribution

I wrote the server, both in terms of network communication, and setting up the framework to process requests, which includes authentication (with salted passwords). I also wrote the https client for the application-side and made the server https compliant. I also had a hand in optimizing the database classes and rewriting some of them to make them more robust and made sure that everyone was on the same page by writing up a standard and consistent JSON API we would use when communicating between server and client.

##### Weak points

I don't think that some of my explanations were clear enough in terms of what I was doing and how to use the frameworks which I set up, so I wrote up the api list and server query how-to guides (in /docs) to illustrate how everything works. I feel like sometimes there were times where I didn't show enough initiative when I should have done, but I hope that I have learned my lessons and that for future projects I will take more responsibility when I need to.

##### Strong points

I made my code as robust and universal as possible, which paid off immensely once we began adding functionality and actually making the whole thing work. I also made sure to thoroughly test (the parts that could be tested) my code and made sure that others knew how to do it as well (once they started writing within my server query framework)

#### 4.5 - Vanessa Timmer

##### Contribution

I worked on the GUI, which consisted of making the visual representation of our application and adding basic functionalities to components of the user interface, so the others could expand on them if necessary or connect them to other parts of the program. Since the instructions were very ambiguous, having to make a user interface from scratch was a daunting task at first but as the project went on and we got a clearer understanding of how we wanted our application to work it became easier to envision how everything should look as well.

##### Weak points

I could've spread out my work better instead of doing most of it on Fridays and in the weekend, not only for myself but also for those who needed me to finish something before they could continue. I also feel like at times I could've shown more initiative, especially at the beginning of the project.

##### Strong points

One of my goals when designing the GUI was to make a program that looks legitimate and not like something that would install a virus on your computer. I think I've succeeded in not only this, but also in designing an application of which the functionalities should be clear to even the most technology illiterate people. I'm also satisfied with my participation during the weekly sessions. In the readme I mentioned that balancing my introversion with group work is something I struggle with but for this project I was able to share my thoughts and comment on other members' contributions.



## Section 5: Value Sensitive Design

### Design for Environmentalists

The main concept of our project is originally aimed at non-environmentalists. But we need to take into account that environmentalists are actually the ones who can influence our program the most. This is because all our calculations are based on data that is inspired by their perseverance in making this planet a better place to live on. They are the ones who encourage and motivate researchers to keep investing time, money and effort into

The main focus to improve our application would be retrieving the data that is used to calculate the CO<sub>2</sub>-emissions. As for right now these calculations are just bare estimations and speculations. To polish these calculations some actual research will have to be done by us, either looking for the more specific data ourselves or looking for someone who has done some research about this. We have also taken far too few factors into account. The amount of CO<sub>2</sub> that is saved by eating a vegetarian meal is not a constant but rather depends on many things such as, what are you replacing your meat with and in what season are you eating your vegetarian meal in. Leaving out all these factors might not be relevant all these non-environmentalists, but I can imagine that calculations with large errors might lead to conflicts with actual environmentalists since the number do not represent the environment correctly. The worst-case scenario would be that an ignorant client of our application thinks he is saving up more CO<sub>2</sub> than he actually is.

Improving our sources of information and data can have many degrees of improvement. A minor improvement would be to simply extend our research in data and add some of these extra factors to the calculations. By doing this more extensive research we might be able to narrow down our error in CO<sub>2</sub> calculation. This research can either be done online. Reading articles and academic papers is a good way to start. The first page after googling 'reduce co<sub>2</sub> footprints' already shows some very promising articles and even a TedTalk on how to "cut your carbon footprint by 60%".

For a bigger improvement we will need to dig deeper. We will have to confront an actual environmentalist and/or researcher. They can give us insight on how far the problem branches out and where we should stop finetuning our calculations. These researchers can provide us more accurate data specifically for our application. The Georgetown University seems to have made some progress into the research of the 'Carbon Footprint'. Also, the Dutch government agency RVO (Rijksdienst Voor Ondernemend Nederland) has a good overview of organizations who are doing research on sustainability and innovation. By digging deeper and trying to truly grasp this global issue we can make our application more accurate.

We must not forget the influence we can have on non-users as well. Yes, our application can even change the lifestyle of someone who will never be in touch with our application. Imagine 2 friends sitting at a bar. Friend X has been using our application whereas friend Y has never even heard about it. Friend X is telling Y about how he is eating vegetarian at least once a week and goes to work by bike. He has been saving roughly 60kg of CO<sub>2</sub> a month. Friend Y is surprised that such a small change in lifestyle can have such a big impact and he decides to change his lifestyle as well. Of course, this is a fictional story and unlikely to happen exactly like that. But the main point is that our application can influence many people, even if they will not use it.

Our application is still very barebones if you think about all the additional features that can be added. And because it is still in its alpha version there is a lot of room for improvement as well. But no matter what we will change to our application, we will always have to take all our stakeholders into account.