GoGreen Report

Computer Science and Engineering
CSE1105 OOP Project
Group 96

April 4, 2019

Members:

- Eliott Afriat 4789393
- Paula Iacoban 4853008
- Ioanna Nika 4788311
- Rico van Leeuven 4779029
- Anoek Winkelman 4571126

1 Preface

This report has been written for the course ”CSE1105 Object-Oriented Programming Project”, which is part of the study Computer Science & Engineering at the University of Technology in Delft.

In the third quarter of the first year we have produced an app to calculate how much CO2 is saved by changing ordinary tasks. The major goals were improving our java, database and client-server knowledge.

The report has been written to describe the process of making the app and the app itself. Features, graphical design, the set-up of the app and ethical decisions will be discussed. Furthermore individual feedback is given by all members of the team.

We would like to thank our professor A. Zaidman, his co-instructor S. van den Oever and the TAs M. Steenbergen and W. Bubberman for helping us complete this project.

2 Summary

Contents

3 Introduction

The desktop application “Go Green” has as ultimate goal, the integration of the environmental thinking into the everyday of an average human. “Go Green” calculates the carbon dioxide saved by the user when performing actions such as buying food that was produced locally, installing solar panels, eating specific meals, lowering the, home temperature, cycling or taking public transportation instead of using the car. All those features aim to warn the user about their habits in order to make him improve and adopt a more eco-friendly routine. ”Go Green” aims to leave a green footprint on the earth.

The goal of this report is to describe the process of making this app and the experiences of the team members during their (first) contact with a project like this. The report is divided into three major subjects. The first part is a description of the product in chapter 4 and a description of the process in chapter 5.

The second part is reflection on both product and process in chapter 6 and individual feedback in chapter 7.

The third part is the ethical discussion about computer science and how that discussion relates to this project in chapter 8.

4 Product

In terms of architecture, there are three main interconnected sections: the client, the server and the database. On top of these the GUI, features, user authentication and authorization are built.

4.1 Client-Server interaction

Spring & Json

4.2 Data management

Connected to the Server. Hibernate. Authentication.

4.3 Features

- **(Vegetarian) meal**
  The user enters the quantity for each of the ingredients present in the meal. Based on this input the CO2 consumed by the specific meal is calculated and compared to the average dutch meal. This is done to derive the CO2 saved and add it to the score. If the CO2 consumed by user per meal is greater than the dutch average meal, then 0 grams of CO2 is saved.

- **Local produce**
  This feature has not yet been added to the app.

- **Take the bike & Take public transportation**
  The user enters the distance traveled in kilometres with a certain vehicle. The input is used to calculate the CO2 consumed by the vehicle. This amount is subtracted from the CO2 produced by the same distance traveled by car. This result is the amount of CO2 measured in grams.

- **Home temperature**
  For this feature the temperature in one's home is used to calculate saved CO2. As input the monthly energy consumed from the utility bill is taken. The surface area of the house and the user's heating system (either electric, natural gas or heating oil) are also asked for.

  From the total amount of energy, 29% is assumed to be the energy used exclusively for heating. This result is subtracted from the average energy required to heat 1 square meter times the surface area. Afterwards, the saved energy is multiplied by the CO2 consumed by 1 KWh of energy according to the heating system the user has. If the energy consumed is greater than the average required energy to heat the house, then 0 grams of CO2 is saved.

- **Solar panels**
  As input the user gives the amount of solar panels they have installed on the roof. This is used to calculate the amount of CO2 saved per day. Every time the user logs into the app a new request is made to check if the time since last logon is more than a 24 hours. If so the saved CO2 is added to user's score.

- **Add a friend**
  On the "Find opponent" page the user can add a friend, searching by a (unique) username. If the friend request is successful a bar chart with the CO2 score of the two users is displayed. If the user searched for does not exist an error message is displayed.
  The user can display only one user's statistics at a time, but can change this easily by adding a new friend. This will remove the previous friends' statistics and the new friends'.

- **Gamification** Badges are given to users if they have certain streaks. For example if the user has been eating vegetarian meals for a week, they will get a copper medal. This will become a silver medal after 2 weeks and a golden medal after 3 weeks.
  A similar thing has been done for the transportation feature; after traveling a certain amount of kilometres, the user will get medals.
  Solar panels are also used for gamification. A setting has been added to stimulate the user to log in every day if they want to gain the score for using solar panels.
  To stimulate the user's competitive streak an "add friend" option has been added. This way the user can compete with their friend to see who has the most CO2 saved.

5 Process

First the team explored the possibilities of JavaFX to make the GUI but later decided to use Scene Builder. There was also a discussion about whether to use Jersey or Spring and later on Spring was chosen.

For gamification the team considered adding more game-like aspects, such as "magic" cards. These would force the user's opponent to perform certain eco-friendly actions to gain points. This would also make the game more interesting graphics-wise. This idea was rejected having not enough time to implement it.

It was decided that instead of having only the vegetarian/not-vegetarian meal option would be more interesting if the user could choose from more options, such as different kinds of meats. The rest of the features were implemented fairly quickly in a similar way.

There were some problems with authentication and authorization. This took a little longer to implement than planned. Furthermore, the database was also troubling. There were parts of try-catch methods that did not want to be tested and some parts had to be rewritten as the original plan did not pan out. Eventually all problems were fixed by the team without too much trouble.

6 Reflection

6.1 Product

Reflecting on the product reveals plenty of ways that lead to improvement and further development of the application. More features could be implemented to illustrate eco-friendly activities like *plant a tree*, as well as adding an android version to allow the user access on mobile devices.
In addition, designing a more complex way of gamification would make the app more engaging. This could offer the user the option to design an avatar or a fighter, which would result in an interesting visual representation. One idea for gamification was introducing card game elements like playing cards with mana points, damage points. On the back-end of things, the connections could be more organized and efficient and in terms of security randomized JWT could be used.

6.2 Process

On the whole, the project has been a very positive experience and it was hard finding things to improve. Gitlab could have been utilised better by using milestones and making releases earlier on. Setting clearly defined weekly goals for the team would have also been a good strategy to implement.

6.3 Course

The course was well organized, especially in terms of deadlines, which helped us research, prioritize, and divide work. The advice and support received from our TA was very useful as well as the resources from brightspace, although some links about maven would have been welcomed. Moreover, the rubric provided a good overview of the step by step process we had to follow and how the points are awarded. Although, the bonus feature section is a bit vague in terms of how complex should the feature be in order to score full or partial points. On the other hand, having more freedom in the conceptual design of the app would have been more exciting, like being able to choose which features to implement.
To sum up, we learned plenty of aspects, not just about programming and java, but also about how to communicate and work efficiently as team and how to make proper use of the tools and resources that we have available.

7 Individual feedback

7.1 Eliott Afriat

During this project, I mainly focused on the database, meaning I designed it and implemented it on the server. My main concern while designing it was to make it as easy as possible for my teammates to interact with the database, and as such tried to make all interactions necessitating a single method call. I made an error early on in not looking at spring when my team started to talk about it, because it would have made my life a lot easier, instead without spring I opted for an implementation of Hibernate which turned out to be more hassle then it was worth. This led me to waste a lot of time trying to redevelop the system half way through the project and ultimately giving up due to a few errors I couldn't resolve the fact that it would have required my team to change their code substantially.
At the beginning of the project, I decided that I would work on my communication skills, because it is a skill I knew was important, and that I was lacking in. I feel that my team side communication was on the whole pretty good, responding to the chat fairly quickly, and expressing my thoughts more adequately than I was expecting at the beginning of the project, although at the beginning of the project I was still in the habit of cutting people off during the meetings, but I feel I have improved on that. Righting this report has ultimately proven to be a bigger challenge for me. I have also volunteered to present the presentation, as that has also been a weak spot for me.

7.2 Paula Iacoban

At the beginning of the project I was researching client-server communications, in order to understand how it works. I tried to make a simple login page to get to know the terminology and functionality of JavaFX. Shortly after we switched however to Scene Builder for simplicity purposes.
Later on, I started implementing the features and looked into possible ways of calculating the Co2 saved by user when performing certain actions. Using the data found in articles on the web, we discussed and decided on the best implementation for each feature.

As an overview: me and Anoek worked on the vegetarian meal and we calculate the Co2 saved according to the quantity of the ingredients present in meal entered by the user. Then, I programmed the calculator for the transportation features according to the distance traveled and home temperature according to the energy consumed. During this process I also made sure that everything I worked on had 99-100% test coverage and that checkstyle was free of errors. Afterwards, I would commit a screenshot of the checkstyle to Git.

Regarding the personal development side, I could say that I have learned plenty of aspects about programming and team development as well as how an app is built from scratch. I realized how important Git and maven are. I also learned the importance of libraries, frameworks and other tools available. All of these gave a significant boost into our work and increased our productivity. Moreover, sprint planning and scrum are great organization tools that gave clarity into the step by step app building process. This also helped me improve my organization and time management skills. Another thing I learned is that communication plays an important part as well. It is important to be explicit when assigning tasks to avoid duplicate work. Overall, I am more knowledgeable about java, more organized and I express my ideas more clearly, but I still need to work on this as there is room for improvement.

7.3 Ioanna Nika

The project gave me the opportunity to get involved and learn how the server-client communication works. Since the beginning of the project I was responsible for the client communication (requests) with the server. I also got involved in the authentication and learned how to work with tokens in order to identify a user. Moreover, I learned how a framework can help you achieve your goal easier and faster.

I was responsible for the graphical user interface. I researched and worked with the frameworks and tools: JavaFX, Scene Builder and JFoenix. First, I made a basic log-in screen in cooperation with Paula using JavaFX. After a while I switched to Scene Builder as it was easier to create more beautiful screens and add functionality to the components of the graphical user interface. Then, on top of those I used JFoenix as it gives you modern options to beautify the GUI.

Furthermore I researched the framework Mockito and tested some of the classes of the client with it. I achieved 100% branch coverage for my code (package client) and 90-100% line coverage. Check style was free of errors. In addition, I became conscious of the fact that dependencies play an important role in programming and that tools like Maven can be really useful and should be used. We also used Jacoco library in order to check our test coverage percentages. Finally, I realized how useful Git is when it comes to a group project like this.

In what concerns the personal development side, I can clearly say that I learned a lot about Java and how to do research on my own. I also learned how an application is build and all the basic steps and practices namely: sprint plan, sprint review, scrum. These are all necessary for a team in order to be organized. I've also realized the importance of teamwork. Each member of the group needs to trust, respect, and be patient, so each member can flourish.

To sum up the project taught me a lot about Java, it made me feel confident that I can figure out how to do certain tasks on my own and realize how to work and organize tasks in a team.

7.4 Rico van Leeuwen

My main responsibility from the start of the project was to work on the server side of the server-client communication. First Anoek and I had to research the frameworks that were available for handling the requests that were to be made by the client. After some research we ended up on using spring with a tomcat web server and my task was to make the server respond to requests made by the client using the spring framework.

In between doing that I also configured maven to have our project structured as a multi-dependency project and I set up the automated Gitlab tests such that it immediately fails if there are any checkstyle warnings.

During the last couple of weeks before the last demo I was tasked to finish what Anoek had done on the authentication and had to work closely with Ioanna on also getting it to work on the client side.

I feel like I learned a bunch about working in a team and I feel like my communication skills improved over the span of this project. In the beginning for example I didn't communicate enough with Anoek and was so used to working alone on projects that I started doing research on things she was supposed to do. When I realized this, I immediately tried to prevent it from happening later on.

Learning how to use maven was pretty nice once I figured out how it worked. I won't ever use jacoco and checkstyle in a multi-dependency project ever though, as I feel like I shouldn't have wasted so much time on trying to set those up properly. Furthermore, I really enjoyed working together with the rest of the team and I feel like a learned a lot about working on a programming project in a team.

7.5 Anoek Winkelman

At the start of the project I was tasked together with Rico to set up the server side. It quickly became clear to me that trying to help with the authentication and authorization was more in everyone's benefit. I chose Spring over Jersey for our application because of the better documentation. It seemed to me like Jersey was meant for people who had more experience setting up projects like ours. Then I made a set-up for the authorization and authentication and worked with Paula to make vegetarian meal feature. Authentication was frustrating to implement using spring-security. It kind of felt like for every question I answered ten more came up. In the end Rico and Ioanna finished authentication while I focused on solar panels.

I have worked a lot in groups like this before, so I don't think I have learnt a lot about teamwork. I had never worked before with Git in this manner even though I had used it. It was really interesting to try figure out the functions and we haven't even used half. Learning about databases was very interesting since we have IDM this quarter. Maven was also something I struggled with in the beginning. Over all I really enjoyed this experience. The team worked really well together and it was really nice to see a very practical application of what we are studying.

8 Value Sensitive Design

Functionality and goal of the Go Green application.
The main functionality of the desk application "Go Green" is to bring awareness to the user. The app is meant to highlight ones actions can influence environmental problems.

Stakeholders
To begin with, the user of the application is one of the main stakeholders. The app calculates the carbon dioxide saved by the user and compares it with the saved carbon dioxide saved by a "friend" of user. In order to beat their friend, the user needs to include more eco-friendly actions in their daily life. Furthermore, the user will be rewarded from the application if they hold up the set standards.

Certainly, the environment is an important stakeholder. Since the application is designed to prompt users to do environmentally safe actions, it follows that it will hopefully have a remarkable impact on the environment. The idea of "Go Green" is to eliminate the carbon dioxide produced by ordinary people. It's sadly true that Carbon Dioxide levels on Earth are alarming. The weakness of humanity to control the Earth's temperature, climate changes, marine organisms that are threatened are just some examples of the consequences of what unnatural levels of carbon dioxide can cause.

In addition, companies that produce and sell solar panels, the producers of vegetarian and generally eco-friendly products, bike sellers as well as the local community (locals) will benefit from the application. The features of the application could be really advantageous for those who support those environmentally friendly actions. "Go Green" will be beneficial for all those stakeholders from an economical point of view since the demand for their products will be increased. Similarly government will increase its income as more people will choose public transportation for their displacement. However, organizations or companies responsible for producing electricity by non-renewable energy resources will be influenced economically as well, that is their income will decrease.

Potential Stakeholders
1) To start with, "Go Green" could be modified in order to be able to be used for educational purposes. The application with some improvements can be a really useful and interesting learning tool and it can be integrated in educational environments. Consider (values): well-being, education,
2) The app could be useful for Non-profit green organizations or the Government as it keeps track of the user's activities. The app can share some of those data in order to help to derive useful statistics about the lifestyle and routine of the habitants. "Go green" with some changes could server this purpose: save more data about the user, make it an android app – GPS. Consider (values): safety value, privacy, trust, innovation (though research - make data available).
3) Companies (maybe eco-friendly) that want to be advertised through the app. Consider (values): economic growth, environmental-friendliness

9 Conclusion

10 References

@miscMilieucentraal, title = Vlees, vis of vega, howpublished = https://www.milieucentraal.nl/milieubewust-eten/vlees-vis-of-vega/, note = Accessed: 2019-04-03

- Co2 values for each ingredient: https://www.milieucentraal.nl/milieubewust-eten/vlees-vis-of-vega/
- https://www.rivm.nl/bibliotheek/rapporten/2016-0200.pdf
- https://www.allesopeenrij.nl/cultuur-2/eten-drinken/hoeveel-en-wat-drinkt-de-nederlander-2/
- http://library.wur.nl/WebQuery/wurpubs/fulltext/464580
- Co2 values for transportation: https://www.delijn.be/en/overdelijn/organisatie/zorgzaam-ondernemen/milieu/co2-uitstoot-voertuigen.html
- Co2 values for bike: https://ecf.com/news-and-events/news/how-much-co2-does-cycling-really-save
- https://www.energystar.gov/products/ask-the-expert/breaking-down-the-typical-utility-bill
- Co2 values for heating systems from https://www.energuide.be/en/questions-answers/how-much-co2-does-my-home-emit/68/
- average energy per square meter from https://www.ovoenergy.com/guides/energy-guides/how-much-heating-energy-do-you-use.html

A Appendix A

Sprint Review	2/24	3/1	3/10
Main Problems: **Problem 1**	People from the group are quitting: Alessandra left the team, Rayan is thinking about leaving	Organization of Git: The team made progress but is still trying to make better use of Git. Ways Git can be organized better and used properly have been discussed.	Deadlines: It took more time for some tasks to be done then expected.
Problem 2	Organization of Git: The team needs to use git in a more organized way, for example when creating issues and branches.	Working with latest version: Some members were working on older versions of our develop branch.	Database testing: The catch branch in a try-catch method is difficult to test. The database side of the team will work hard to improve.
Problem 3			Different implementation: Some tasks were not implemented as the team had discussed.
Adjustments: **Previous Sprint**		Increase Testing effort: progress has been made. Better use of Git: working on it. Redivide workload and assign issues: succeeded.	Testing effort has been increased.
Next Sprint	Increase testing effort. Better use of Git. Redivide workload and assign issues	Better communication. Productivity. Organization	Better communication. Productivity. Organization.

Sprint Review	3/17	3/24
Main Problems:		
Problem 1	Too much focus on demo: Next time there will be more focus on specifying tasks and better dividing of tasks.	Some tasks haven't been finished in time: Other courses took precedence over the OOP-project.
Problem 2	Not enough clear documentation for authentication: Other team members are going to help, so the authentication is finished for the upcoming demo.	
Problem 3		
Adjustments:		
Previous Sprint	Communication and organization have been well executed. The team has been very productive.	Better communication achieved. Organization improved.
Next Sprint	Divide to-do's better.	Increase productivity