GoGreen Report

Computer Science and Engineering
CSE1105 OOP Project
Group 96

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, the TAs M. Steenbergen and W. Bubberman for helping us complete this project.

2 Abstract

Global warming is a serious problem and has to be dealt with accordingly. The application "Go Green" has been created to offer insight on solutions to its users; how do their actions directly affect the environment.
This report has been written to describe the process of making the app, the learning process of each team member and the ethical decisions that were made.

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

The web server of the Client-Server interaction is provided by tomcat and Spring MVC is managing the content that is served to the user. Spring MVC is used because it's widely used and therefore a lot of documentation and examples can be found online which make it more easy to work with compared to alternatives. Tomcat is used because it is included by default in the spring web plugin package and provides everything needed.

For the client to interact with the web server Apache HttpClient is used since there is a lot of documentation available online and it's still being actively maintained.

All data that is included in the clients http request body and all that is returned by the server is converted to json and converted back to a class by jackson as this library is built into spring and thus very easy to use.

4.2 Data management

The database is running on MySQL, and Hibernate is used to map the objects to the database. This is done sub optimally because Spring was not utilised.

There are four tables in the database, user, action_log, continuous_action and badges. As an security measure, the passwords are stored in hashed. All discrete actions give a reward for each time you do them. These are stored in the action table and are linked to the user, continuous_action was designed to support multiple types of discrete actions. At this point of time that is only solar panel. All badges achieved by the users are stored in the database and linked to its user.

Each instance of an Object represents an entity in the database, which is managed by an entity manager. To modify or retrieve a row, most manager methods require the use of a primary key.

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 [n.a., nd, Ocké, 2018, Hoogeveen, 2010, Dagevos et al., 2018]. 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**
  As for the meal feature, the user inputs the quantity for each of the products bought. The Co2 saved is calculated by computing 7% of the Co2 consumed [n.a., 2015].

- **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 [Ketelaars, 2019, Emmen, 2013]. 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 [Bailey, 2016]. This result is subtracted from the average energy required to heat 1 square meter times the surface area [n.a., 2019, n.a., 2009]. 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 designing and implementing the GUI. The decision was made on JavaFX, since is the most widely used library for graphics and is more modern than Swing. Moreover, Scene Builder was a great tool which simplified and boosted the graphics creation process and styling was improved using JFoenix. There was also a discussion about whether to use Jersey or Spring. Later on, Spring was chosen because it provides a clearer documentation and it is easier to use.

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 with. We decided to implement JWTs and store hashed passwords in order to make our app more secure for the user. Although, 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. In addition, implementing the database with Spring instead of Hibernate would have been less complicated, but is was already too late to change. Eventually all problems were fixed by the team without too much trouble.

6 Reflection

6.1 Product

The product could benefit from further development in many ways. 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. The features solar panel and meal could also be improved. Solar panel checks if it has been more than 24 hours since the user last logged on and not if the user logged on the day before. Meal could be improved by only calculating the CO2 saved once a day instead of every time after the user adds a meal.
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 my task was the design and implementation of the database. Easy interaction with the database was my main concern. As such I tried to make all interactions necessitating a single method call. Instead of Spring, I opted for an implementation of Hibernate which turned out to be more hassle then it was worth. This led to trying to redevelop the system half way through the project. Ultimately I had to give up due to a few errors and the substantial change of code the redevelopment brought with it.

At the beginning of the project, I decided that I would work on my communication skills. 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. 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. Writing 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 start of the project I was researching client-server communication and tried to make a simple login page to understand the functionality of JavaFX. Afterwards, we switched however to Scene Builder for simplicity purposes. Later on, I started implementing the features. 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. Then I programmed the calculator for the transportation features, home temperature and local produce. During this process I also made sure that everything I worked on had 100% branch coverage and that checkstyle was free of errors.

Regarding the personal development side, I could say that I have learned plenty about programming and team development as well as how an app is built. I realized how useful Git and maven are. I also learned the importance of libraries, frameworks and other tools available.

Sprint planning and scrum are great tools that helped me improve my organization and time management skills. Another thing I learned is the importance of communication to avoid duplicate work. Overall, I am more knowledgeable about java, more organized and I express my ideas more clearly. However there is always room for improvement.

7.3 Ioanna Nika

The project gave me the opportunity to learn about server-client communication. Since the beginning was responsible for the client communication (requests) with the server. I also worked on authentication.Moreover, I learned how a framework can help you achieve your goal easier and faster.

I was also responsible for the GUI. 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 implement. 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. In addition, I realized the importance of dependencies and the usefulness of Maven and how useful Git is for group projects.

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 how sprint plan, sprint review and scrum are all necessary for good organization. I've also realized the importance of teamwork. Everyone needs to trust each other so they can all reach their potential.

To sum up the project 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 researched the frameworks that were available for handling the requests that were to be made by the client. We ended up on using Spring with a Tomcat web server. My task was to make the server respond to requests made by the client.

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 authentication and getting it to work on the client side together with Ioanna.

I feel like I learned a lot. In the beginning for example my communication wasn't great. I was used to working alone on projects, which is why I accidentally did research Anoek was supposed to do.

Once I figured out how maven works it was really nice to use. I won't ever use jacoco and checkstyle in a multi-dependency project ever though, as it takes a lot of time trying to set those up properly. In conclusion, I really enjoyed working together with the others 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 app "Go Green" has as goal making the user more aware of the consequences of their actions on the environment. Awareness of problematical habits is the first step to adopt a more eco-friendly routine. "Go Green" aims to leave a green footprint on the earth.

Stakeholders
To begin with, the user of the application is one of the main stakeholders."Go Green" expects that the user by interacting with the app for some time period, will make a habit of executing the daily features.

Certainly, the environment is an important stakeholder. Since the application is designed to prompt users to do environmentally safe actions, it follows that these actions impact the environment in a positive way. Companies that invest in eco-friendly products will therefore benefit from the application, since the demand for their products will be increased. Similarly, the income of government will increase as more people will choose public transportation.

Potential Stakeholders - Values - Potential sources

"Go Green" could be modified in order to be able to be used for educational purposes. Specifically the app can be programmed to provide to the user information about the carbon cycle, whilst educating students about sustainability. "Go Green" could include quizzes and games in which the user will be tested in applying in a simulated environment. Teachers can design the content of the quizzes and decide on which topics the information displayed by the app should be focused on, according to the curriculum.

The app could be useful for non-profit eco-friendly organizations or the government as it keeps track of the user's activities. Useful statistics about the lifestyle and routine of its users can be shared. "Go Green" can be transferred to a mobile app, so it can implement features such us pedometer. This will enable to track the distance the user walked accurately. Concurrently, eco-friendly organizations can use the data in order to understand in which areas people are lagging behind, so they can sensitize and maybe promote or even sponsor those actions.

The features above may cause safety and privacy issues. It is necessary that the application protect the data and the privacy of the users. The application should be trustworthy. Concomitantly, it's important for the application to share some of its data, since they can lead to innovations. For to achieve both, the development team can be enriched with data scientists and specialist in data privacy and security. Lawyers can be also hired in order to ensure that the application handles data in a legal way. The application can also promise to its users that it shares their data in the absence of their identity.

Additionally, eco-friendly companies that want to be advertised through the app either for being environmentally friendly or for producing and selling relevant products might be interested in the app. The application can include all those relevant advertisements in its GUI. The development team of the app can also negotiate and make agreements with such companies in order to offer to the user real life rewards such as discount coupons for relevant products.

As the app will promote eco-friendly companies, firstly not only it will contribute to their economic growth but also it will thrust the user to buy green products. Finally in order to bring this idea into realisation specialist in marketing, finances but also psychologists should be hired in order to take care of the advertisements and display them in a way that is accepted by the user.

9 Conclusion

The application is meant to bring insight to its users on their habits and maybe change their behaviours. Motivators such as gamification have been used, but could be better exploited by making the game more graphically interesting. The team members overall had a positive experience during the project. Improvement could be made by setting better defined weekly goals and better communication about implementation of current tasks. To better the ethics of the application, stricter security could be used to ensure the privacy of the user.

References

[Bailey, 2016] Bailey, A. (2016). Breaking down the typical utility bill.

[Dagevos et al., 2018] Dagevos, H., Verhoog, D., van Horne, P., and Hoste, R. (2018). Vleesconsumptie per hoofd van de bevolking in nederland.

[Emmen, 2013] Emmen, R. (2013). how much co2 does cycling really save?

[Hoogeveen, 2010] Hoogeveen, P. (2010). Hoeveel en wat drinkt de nederlander.

[Ketelaars, 2019] Ketelaars, J. (2019). Co2-emissions of vehicles.

[n.a., 2009] n.a. (2009). How much energy do you use to heat your home?

[n.a., 2015] n.a. (2015). Eat your way to a smaller carbon footprint.

[n.a., 2019] n.a. (2019). How much co2 does my home emit?

[n.a., nd] n.a. (n.d.). Vlees, vis of vega.

[Ocké, 2018] Ocké, M.C., T. I. G. M. M. M. T. E. H. N. (2018). Wat ligt er op ons bord?

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