

Go Green

CSE1105 OOP Project 2018/19 Q3

Group - 95



Group members

Edin Avdić	- 4711793
Mark Basting	- 4904923
Kevin Landegent	- 4364201
Filip Marchidan	- 4866657
Giacomo Mazzola	- 4872673
Jordy van der Tang	- 4213882
Jim Vos	- 4922905

Index

General	3
Design Decisions	3
Users	4
Activities	4
CO ₂ savings	4
Attributes	5
Solar panels	5
Achievements	5
Graphical User Interface	5
Spring	6
Points for Improvement	7
Individual Feedback	8
Value Sensitive Design	10
Appendix A	11
Appendix B	14
A conceptual ER-diagram of the database of the application	14
The database schema.sql	15
The definition of the events that are running daily on the database	16

General

When it came to working together on this group project we were definitely all able to get along sufficiently on a personal level which is an important basis to get the group project running properly. On a professional level we could collaborate as a well organized team . When we drafted this project of what it should look like, everyone was able to give their input and be heard, at the beginning and throughout the project.

Even though the professional collaboration went well, there were complications so there is room for improvement. Communication was not always working well, we had a whatsapp group and a mattermost group for communication with the TA. Also, we had one or two meetings per week above the mandatory meetings on Monday. The face to face meetings were always pleasant, basically any troubles we had ended being hashed out after the meeting. This was not the same for online communication. This went slower and did not get the same input from everyone. This was eventually discussed in the face to face meetings and we did see some improvement.

What also went wrong was the input of everyone. The input varied from people who were not able to deliver on agreements in the sprint plan to people who did more than was asked. This was something not immediately visible at the start of our project, but it became very obvious later in the project who did more and who did not. The only attempt we could make was to restore balance as much as possible in the final weeks, this we did by dividing the workload fairly so everyone learned as much as possible.

There were some things that went right as well, for example we were always able to meet our deadline, despite any complications. This was possible despite the fact that we sometimes ran into last minute trouble at times.

As for the version control, we found really handy using Gitlab and Sourcetree. Even though at the beginning it wasn't really clear how they work, in the end we learnt how to use them in order to make our work easier. Doing sprint plans and reviews and using tools like Scrum board definitely helped us a lot scheduling and splitting our work. Merge requests allowed us to capture big mistakes before merging, but usually we failed to catch smaller errors.

One of the more important things that we became aware of during project is the complexity of working together in a large group. We have all already worked in projects but most of us never worked in teams of this size. It does come with its pros and cons. Dividing the workload on a bigger scale was one of the benefits we enjoyed from having a big group, but something that was troublesome was keeping the entire group in formation and properly communicating with each other. Code wise, we have all dealt with unfamiliar concepts, we have learned from our own work and research but also from each others work.

Design Decisions

Throughout the project several decisions have been made regarding the structure and way of handling things in the program, in this chapter we will discuss how and why those choices came to be, as well as some of the upsides and downsides.

Users

Users are the key component of this application connecting to most of the other elements; activities, achievements and friends. Because of the nature of these relations, one to many and many to many, these properties needed to be stored in more complex data structures on the java side. We have chosen to use sets to store these, to make sure that it is not possible to accidentally add duplicate values to these relations.

To uniquely identify a user, they have an ID, however their email address is also unique for registration and login purposes. The user class also keeps track through a boolean on whether the user has a solar panel installed or not.

We keep track of the users' total score directly, which is incremented and decremented through triggers in the database every time an activity is added or removed. We have decided to do this due to a large amount of queries that could potentially overload the database as well as slowing down the performance on retrieving scores.

Activities

One of the main components of the program is the activities. 'Doing' these activities, allows the user to keep track of their CO₂ savings. However, there are a wide variety of 'activities' in real life, and it is important that the digital version is able to decently represent all of these while still being relatively easy to work with.

In the end we have settled on using a single activity class that is decently fitted to accommodate all the required types of activities from a data standpoint. In the end the CO₂ savings of an activity are calculated by multiplying the base amount of savings by the unit amount, so kilometers, days or degrees. Ultimately, they are added to the total score of the user.

CO₂ savings

We decided to use CO₂ savings that were as close to reality as possible. So if this app might be published one day, the CO₂ saving are already accurate. To get the CO₂ savings per activity, we looked at the emissions of a normal scenario and subtracted the emissions of the activity that can be chosen to save CO₂ emissions. For example, for bike and public transport the savings are calculated by subtracting the CO₂ emissions of the selected option from the CO₂ emissions of an average car. We based all these savings on an average suspected scenario per activity. For all the data needed for CO₂ emissions per activity we looked at scientific papers which you can find in Appendix A. These scientific papers provided the following data:

App option	Saved gram CO2	Unit
Vegetarian meal	1361	Per day
Public transport	184	Per km
Bike	220	Per km
Solar panels	2107	Per day
Buy local	700	Per day
Turn down thermostat	816	Per day

The CO₂ values are not directly stored in activities but rather normalized by storing them in Activity Types that are loaded into the database from JSON, so easily editable. The activities hold a reference to these Activity Types. This was done so that it is easier to update/balance the score in the future to make the gamification more fair. On the client side these activity types are referred to using enums, which has the advantage that the id's are more human readable, which reduces the chance of choosing the wrong type of activity.

Attributes

Activities contain a small amount of attributes; the type of activity as mentioned above, the amount of the activity (more on this later), the date at which the activity was added, the user the activity belongs to, as well as a unique id to help identify the activity.

The amount attribute of an activity has been designed to be versatile, it is connected to the unit of CO₂ savings as discussed above. If the user wants to add more vegetarian meals they will have to add more activities. However, for activities such as public transport and taking the bike, the amount indicates the number of km travelled, and for solar panels it indicates the amount of days. For lowering the temperature the amount indicates the difference from 21 degrees, as this was the base average we found.

Because of this it will be straightforward to expand the amount of activities with similarly functioning activities in the future. If an activity can be quantified into CO₂ savings per unit it can be implemented. However, activities that are not easily quantified will be more difficult to add.

Solar panels

The solar panel activity is a demonstration of being able to utilize the basic activity structure in a more complex way. Because solar panels are not something that you either do or don't on a daily basis we made the choice to not require the user to select this every day. Instead, a daily event will run on the database which will find all users and increase their score and solar panel amount.

Achievements

The implementation of achievements is currently simple. An achievement has a value, name and id. We decided on two types of achievements with this data structure: The first type are Achievements that are earned for completing x amount of a certain activity, these achievements contribute to the user's total score. The other type of achievements, are achievements that get triggered when having earned a certain amount of total score. For example, a bronze badge at 100.000 points or a gold badge at 1.000.000 points. Since these achievements are earned based on a certain amount of points, they are not worth any points themselves, but just to show off. With this system it is easy to add new achievements to the database, however there are some downsides which will be discussed in points for improvement.

Graphical User Interface

One of the most important things of the application is, of course, the Graphical User interface. With JavaFX, the library we used to make the user interface, it is very simple to connect the Client and the GUI. JavaFX also allows for separate Cascading Style Sheets (CSS) and has support for signature gestures on mobile ¹. In combination with the Scene Builder from Gluon, it was possible to design a modern Java application with all the functionalities we needed. It is simple to integrate Gluon into IntelliJ, next to that, Gluon also gives better visual feedback while designing the user-interface. This allows for a faster workflow.

¹ educba, *Javafx vs Swing*, unknown <<https://www.educba.com/javaafx-vs-swing>>[accessed

The GUI must be easy to work with and has to be self-explanatory. That is why we decided to separate the different functions as much as possible. We achieved this by having different tabs, for activities, achievements etc. This allows for a smoother user experience.

The UI on the activity page is a reflection of the design of the activities in the background. The slider for some activities indicate that the user has to specify an amount. The solar panel is simply a checkbox that only has to be checked once (with text that indicates the number of days the checkbox is checked) and is designed to reflect to the user that this is not a recurring activity, but something they only have to indicate once.

We separate most parts of the UI to make it easier for ourselves. But we later found it that it sometimes complicated things as well, for example when dynamically changing the total score. We tried to separate the design as much as possible by using a CSS file with all the colors, this way we can easily change up the design if we want to.

On the login page, we stayed with the nature-type design and alerts when the user doesn't have an account yet or types in the wrong password. After users log in, they get redirected to the home screen, where they can switch tabs.

We also wanted to make the achievement page look appealing since achievements can really encourage users to choose CO₂ saving options. So we decided to make badges for every achievement. We made all badges in the same shape and chose color schemes and icons that should make them easily recognizable. We chose to set the opacity of the not yet achieved badges to 10% so that the user can easily see how many and which badges can be achieved.

Spring

For the first deadline we were running late, we technically had one week to get a functional client + server setup working. Because we first ran into two dead ends, first trying to use sockets, and then trying to implement a HTTPS java server directly, we did not have a lot of time left before the deadline. Due to this the choice for using Spring does not stem from comparison to other options, but judging Spring on it's own merits. There was a lot of documentation to be found on spring and it became apparent that there was information to be found on how to use and set Spring up. It makes certain tasks such as setting up endpoints simple and leaves a clear overview with the annotations that are used. Next to that it comes with several 'ready made' solutions, such as using Spring Security to help with user authentication.

Spring also offers ways to connect to databases, so we knew at this point that we knew we were not choosing a solution which would get us stuck in a difficult situation later on in the project. Spring's implementation provides standard access, such as retrieve by primary key but also allows users to create custom queries to get more precise information from the database.

Points for Improvement

Although we think that our application is well made, there is always room for improvement. In this chapter we will discuss what could have been done better or added.

One of the things we think that we could have done better is the way we implemented the achievements. From the user side everything works fine, but we made it rather difficult to add new achievements. This is caused by a lot of hardcoding in the server code when checking for the conditions of the achievements. We would rather be able to implement them so the conditions are directly linked to the achievements and the server can loop through achievements to see if the conditions have been met.

The speed of communication between the server and client is something we could improve on, to better the experience of the user. The problem with optimizing the speed is that we are using a tiny database on a raspberry pi. For a smoother experience we must improve our hardware which was not in our interest during this project.

The GUI is in a good state but could use some improvements in the smaller details. Take the buttons for example; we would like to make more use of some pictograms instead of text, this would make the GUI just a bit cleaner. The scaling of our application is something we could improve on too, currently we are only able to minimize our app into the taskbar. For improvement we would like our application to be resizable by the user.

Next to improving our current functionalities we also wanted to add some functionalities, but the lack of time made us choose not to do so. The user has little freedom to personalize their profile. This could be improved by implementing a customizable user profile, where you can add your profile picture, a description of yourself or photos of your green activities. Also we would have liked to give more information on what your progress in our app meant for the environment. Currently the only feedback you get is the amount of CO₂ in grams and for a lot of people this is just a meaningless number. With some more knowledge on your environmental impact, people might be more inclined to do more green activities.

Working with each other in person we do not think we can improve on much, one point of improvement is that it was sometimes difficult to keep everyone's focus during the Monday meetings.. The main part of communication that was lacking was when we were all at home and tasks were not really clear to everyone, partially because of the meetings. The scrum board was the main solution for this problem, but did not get used to its full potential. We could have stated more information on the scrum board on how we were doing with our progress and what we were changing, this could be done by adding comments at the issue. A lot of this information got lost in little meetings, in our whatsapp group or was not even mentioned until it was finished.

Git is also a tool we did not use perfectly, we did not use master as the most up to date version. This created some problems where not everyone did branch from the most up to date version. When we needed to merge this often gave a lot of merge conflicts. This definitely improved during the project in the end, our use of git became a lot more streamlined.

The course was a really nice experience and felt like a breath of fresh air, although we think there could be some points to improve on. As we already mentioned, we did have some struggles with sourcetree. We think that instead of giving a lecture, it would be more beneficial to let the students do it themselves with guidance from a lecturer. Some more guidance on working with the pipeline, and proper practices such as not pushing database connection information to git would be nice as well.

Individual Feedback

Jordy:

During the project I felt like the scrum board and sprints helped me a lot with my personal goal of being more organized. It allowed me to create an overview of my tasks and see how far along I was. I also managed to plan certain parts of the week for these tasks in my weekly calendar so I knew what to work on and was able to reschedule if I wasn't able to work during the reserved time slot.

During the project I had the impression that I was more active/motivated than some of the group members, and struggled with the lack of response in the group whatsapp at times. This may also have led to me doing some tasks myself rather than trying to get other members to do them. However towards the end of the project the general responsiveness and discussions in the group chat improved and the motivation seemed to improve and in general people seemed to become more involved in the project. I also feel like I was better able to try and ask other people to help/do something, rather than tackle things alone.

In general I did learn to be more patient with people, 'helping' by showing a solution before people have had a chance to try and figure things out for themselves is not really helping, it is preventing them from learning and understanding things properly.

Kevin:

My personal development plan was to improve my coding skills a lot. I thought I was able to improve this a lot but since we already had deadlines very early I didn't had a lot of time to start improving my skills. So I decided to focus on other things like making a design sketch for the app and finding data about the CO₂ savings. After a talk with our mentor, he suggested that I should try pair programming. I tried to help with the UI section of the project since that seemed like the most easiest part for me to work on. In the last weeks I was assigned the achievement functions. And i was doing the UI part. At the end I now know how to set up an css with javafx for java applications. And in the last week I was even able to work on the java part of the UI thanks to Mark helping my pair program. Looking back this project definitely has improved my coding skills. Not so much that I hoped for but I also learned a lot working with a version control program. And i also gained a lot of experience working in a big group because all my previous projects where in groups of four or smaller. And I learned with such a big group somethings like making a planning and dividing tasks are really important and need to be clear.

Mark:

At the beginning of the project, I was a bit overwhelmed with the amount of work that had to be done, but I tried to focus on different small task that had to be done. The scrum board personally helped me a lot with this.

I definitely should have been more active in the group meetings and chat, but this improved once we were a few weeks in the project. I still feel like I should have offered to do more tasks, but I didn't because I thought that I would slow the group down. Just like I said in my personal development plan. The way we worked in separate groups is something that definitely helped. This way we could both learn from each other. During the project, I also pushed myself to step out of my comfort zone and just try to do tasks in which I had no expertise. This was pretty hard for me since I tend to procrastinate when I don't really know what to do.

Something we could have done better as a group is planning. Definitely at the beginning of the project. This led to our productivity being lower in the first weeks. This improved over time, but I feel like we could have done a better job. Overall, I am pretty proud of the work we have done.

Jim:

My personal experience with making this project is all round very good. Not only did I enjoy making the application and working in a team, I also did learn a lot in a very nice manner. Never have I ever worked in a group of a size bigger than 3 and I did not have a lot of coding skills. The lack of experience on these parts did make it a bit harder for me in the beginning. This showed itself in my communication which was not always transparent and a slower pace of working on my tasks. During the project I could improve a lot on these weaker points by the help of my teammates. The paired programming was very helpful to better my coding skills and I could adapt my communication on how my teammates have been communicating towards me. One of the main parts where I could help my team was in the GUI part, because I got the time to acquire the knowledge to improve my tasks. The main conflicts we had during this project was the level of communication. Splitting the group into task divided groups was a nice way of working, only merging everything together and solving the problems outside of these main tasks was sometimes a problem for us. We did not always communicate on our tasks and the little but important tasks often got left until the end. This definitely improved along the way, by communicating during the meetings on what was going wrong and could get improved. I think we did a great job and have created a nice application in the end.

Giacomo:

Thinking about my personal development plan I'm glad to say that I made some improvements regarding my weak points: I overcame totally my shyness showing myself really active in the project and proposing a lot of ideas; as for the organisation of the work I still have many steps to take but I have to say that using tools like scrum board really helped me a lot.

During this project I learnt a huge variety of things that I believe will be useful in the future: using git, a framework as spring, designing a database with MySQL, work in a team...

What I liked most of my group is that people didn't work only on a specific task and so we had a chance to grow significantly our knowledge.

Due to a lack of experience of the group members, at the beginning we had many problems using git properly but in the end we improved much.

I think the biggest problem we encountered was a lack of communication between the members of the group that sometimes made us wasting time doing twice the same work.

I put a lot of effort in this project and I'm proud of the result. Obviously it can be improved significantly but taking in count that this is our first project we can be really glad about it.

Filip:

Throughout the whole project I have personally experienced both upsides and downsides. To begin with, the idea of a team project I thought it brilliant because not only it prepares you for future working environments, but it also helps you to improve your collaboration skills as you learn how to help each other and divide tasks in order to accomplish a target or solve a problem.

One of my strong points was that I was always able to help my team and contribute whenever there was need of such or even taking the leadership at certain times when needed. Another strong point I had during the project is that I was able to provide quality code always before deadlines that were either set by the course itself or our personal internal deadlines.

However, I also recognized weak points such as my bad habit of not creating separate branches when working with git or the SourceTree application which could have been potentially disastrous. However, as I was reaching the end of the project, I started to correct this bad habit by creating separate branches for features or testing.

Another weak point I have noticed is that I spent quite some time at understanding the insights of the spring framework which delayed to some extent my performance. During the project overall, I have not experienced any conflicts whatsoever with the other team members, as we commonly agreed to discuss any issues together and not act on our own. However, the only conflicts that we have encountered were “bad” commits on certain branches which took only a bit of collaboration to solve them and it all went alright with no issues. To conclude with, I have encountered throughout the whole project both strong and weak aspects of myself and managed to overcome my weak points by working and improving on them as well as improving on my collaborative skills.

Edin:

The group project was quite a ride, it will say that I have quite some experience doing group projects, but nothing compares to this. I did feel like I have learned a lot of code from my own research but also from the others. I will say that I have learned the most from pair programming, so I’m very thankful for that and will continue using that in the future. But during pair programming a lot of my strength come upfront like taking initiative in the form and also discussing what could work best.

The only prior experience I have had during this project was with mostly web services like HTML and not so much with object oriented programming. I think during this project and during this year I have heavily underestimated the importance and complication of OOP, which led to me trying to figure as much out on my own and not reach out to others as much as I should have.

I compared it to HTML and other web languages and thought this is just another language to learn, which led to me sometimes not delivering as much as I should have during this project and partially burdening my teammates which was never my intention. I think I have learned that before I start working with other people, I should be better able to estimate what the work contains and the knowledge it requires. This is definitely something important to reflect on future group projects.

Value Sensitive Design

When we designed our application, we tried to think and realize our product from an ethical point of view as well. Our application is called “Go Green” and it’s main goal is to reduce people’s impact on the environment. How can it do so? Well, it offers the user a variety of actions that they can do, to produce less CO₂ and to be able to calculate the user’s impact as well as their improvements.

Furthermore, we have implemented a competition between the user and their friends in order to stimulate them to put a lot of effort in their activities. Last but not least, we also thought about a variety of achievements and badges that a user can earn if they complete a certain amount of activities.

One non obvious stakeholder that we thought could be strongly influenced by our application is the food economy. In fact, we are trying to make people eat more vegetarian food instead of meat and to point them towards local products. If it was the case, it would mean that the local vegetable farmers’ economy would increase significantly, but it will have a negative impact on cow and pig farmers.

Appendix A

Option to choose a vegetarian meal

(Rosi, 2017, p. 6105)

Rosi, F. D. S. F. D. R. M. B. J. G. D. M. R. A. L. C. R. R. I. D. C. E. F. S. N. N. T. P. P. A. M. (2017). Environmental impact of omnivorous, ovo-lacto-vegetarian, and vegan diet. *Scientific Reports* 7(1), p. 6105.

<https://www.nature.com/articles/s41598-017-06466-8>

Indicator	Food group	Diet type		
		O	VG	V
		N = 51	N = 51	N = 51
Carbon Footprint (g CO ₂ eq./d)	Drinks	430.9 ± 342.9a	299.2 ± 355.3a	325.0 ± 385.0a
	Meat and Fish	1447.2 ± 756.8a	0.0 ± 0.0b	0.0 ± 0.0b
	Other animal-based foods	901.9 ± 363.6a	628.9 ± 465.2b	0.0 ± 0.0c
	Cereals and their derivatives	425.5 ± 110.1b	490.4 ± 133.4ab	548.0 ± 200.7a
	Other vegetable-based foods	503.3 ± 170.1c	995.8 ± 367.5b	1422.5 ± 381.4a
	Sweets and desserts	250.8 ± 125.1a	184.1 ± 109.6b	47.0 ± 44.5c
	Total	3959.3 ± 975.8a	2598.3 ± 619.0b	2336.1 ± 496.8b

Table 1:

- Values are mean ± standard deviation of fifty-one independent measurements. Different letters indicate significantly different values ($P < 0.05$) as calculated by one-way ANOVA with *post hoc* Tukey HSD test among the three diet groups. O, omnivores; VG, ovo-lacto-vegetarians; V, vegans. Drinks: alcoholic beverages, soft drinks, and fruit juices. Meat and Fish: meat and meat products, and fish. Other animal-based foods: eggs, milk and dairy products, and animal fat. Cereals and their derivatives: cereals and their derivatives. Other vegetable-based foods: fruit, vegetables, nuts and dried fruits, legumes, potatoes and other tubers, vegetable alternatives, and vegetable fat. Sweets and desserts: sweets and desserts.

Choosing to go vegetarian for a day saves you : $3959.3 - 2598.3 = 1361$ g CO₂ or

Option to choose public transport

Category	Unit	g CO2/unit (WTW)	g CO2/unit (TTW)	g CO2/unit (WTT)	Explanation
Car	Vehicle kilometer	220	181	39	based on a fuel mixture of 65,5% Petrol, 31,1% Diesel, 3,4% LPG and a car of average weight (ca. 1170 kg). And an average occupancy rate of 1.39 persons per car.
Public transport	Vehicle kilometer	36	25	11	based on 19% bus,, 3% tram, 3% metro, 75% train.
Bike	Vehicle kilometer	0	0	0	

Table 2: transport carbon emission

With a bike you save 220 g CO2/km

With public transport you save $220 - 36 = 184$ g CO2/km

Source: CE Delft, 2014. STREAM personenvervoer 2014

Option to install solar panels

Category	Unit	g CO2/unit (WTW)	g CO2/unit (TTW)	g CO2/unit (WTT)	Explanation
Grey mix	kWh	649	572	77	Based on the dutch energy mix of coal, natural gas and nuclear energy.
Solar panels	kWh	0	0	0	

table 3: emission per electricity group

With solar panels you save 649 g CO2/kWh.

Average daily consumed kWh per person is 3.25 kWh

(<https://www.cbs.nl/nl-nl/achtergrond/2018/14/energieverbruik-van-particuliere-huishoudens>)

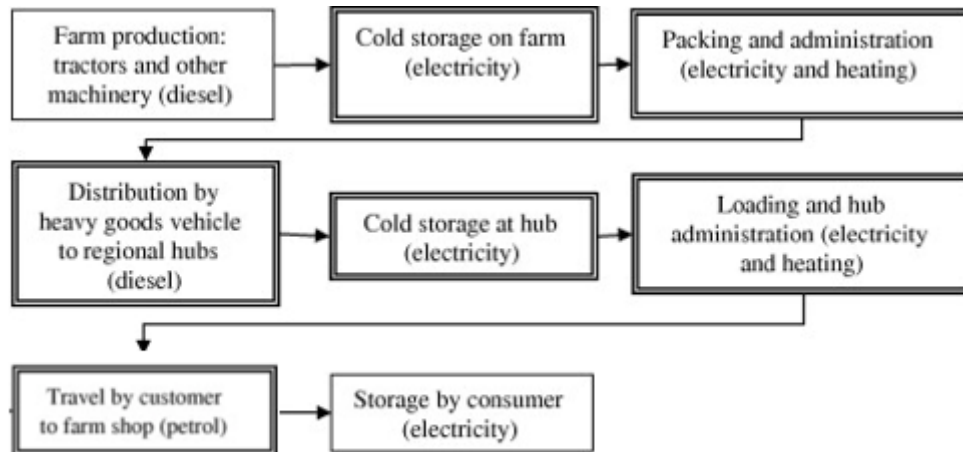
So total daily saved is 2107 g CO2/kWh

Source: Milieucentraal, CE Delft & Stichting Stimular, 2017. co2factor stroomverbruik

<https://www.co2emissiefactoren.nl/co2emissiefactoren/co2-factor-stroomverbruik-20-11-2017/>

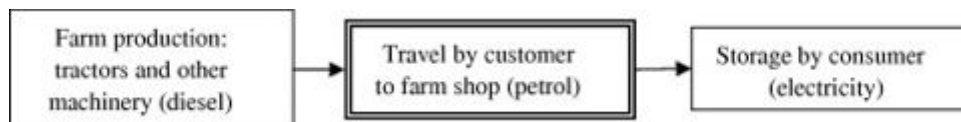
Option to buy local

Flow of product local farm



Source, <i>i</i>	<i>E_i</i> , gCO ₂ /box
Packing, cold storage and administration at farm	300
HGV transport	360
Intermediate cold storage and administration at hub	40

Flow of product large scale supermarket chain



Buying local saves you 700g CO₂

Source: Coley, D., Howard, M., & Winter, M. (2009). Local food, food miles and carbon emissions: A comparison of farm shop and mass distribution approaches. *Food Policy*, 34(2), 150-155.

Option to turn thermostat down 1 degree

Per household you save 1530 kWh/year if you lower your home temperature by 1 degree. So per day this saves you 4.2 kWh. 1Nm³ gas equals 9.8 kWh and 1 Nm³ gas equals 1890g CO₂. So this equals 816 g CO₂

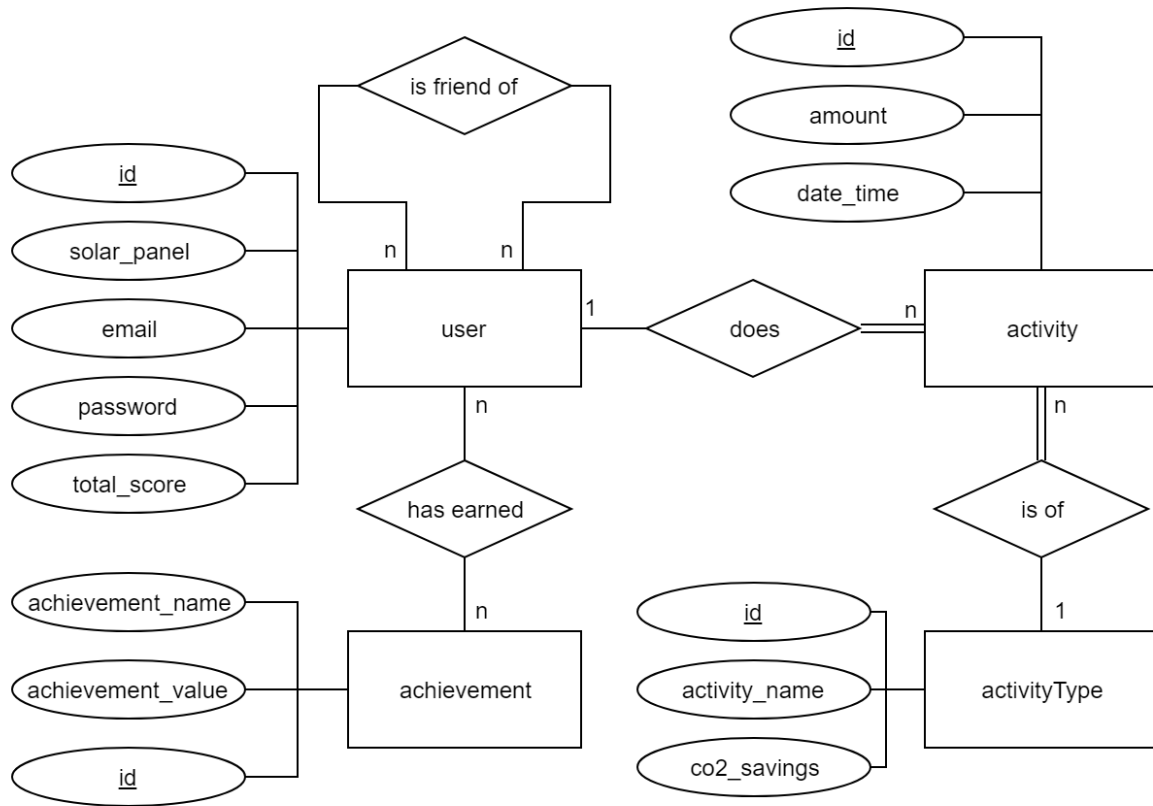
	Unit	g CO ₂
Natural gas	Nm ³	1890

Source: Palmer, J., Terry, N., & Pope, P. (2012). How much energy could be saved by making small changes to everyday household behaviours. *Cambridge Architectural Research*, November.

Milieucentraal, CE Delft & Stichting Stimular, 2017. co2factor Brandstoffen energieopwekking
<https://www.co2emissiefactoren.nl/2017-12-brandstoffen-energieopwekking/>

Appendix B

A conceptual ER-diagram of the database of the application



The database schema.sql

```

create table achievements (id integer not null, achievement_name varchar(255)
not null, achievement_value integer not null, primary key (id)) engine=InnoDB
create table activities (id integer not null, activity_amount integer not
null, activity_type integer not null, date_time varchar(255) not null,
user_id integer, primary key (id)) engine=InnoDB
create table activitytypes (id integer not null, activity_name varchar(255)
not null, co2_savings integer not null, primary key (id)) engine=InnoDB
create table hibernate_sequence (next_val bigint) engine=InnoDB
insert into hibernate_sequence values ( 1 )
insert into hibernate_sequence values ( 1 )
create table users (id integer not null, email varchar(255) not null,
password varchar(255) not null, solar_panel bit, totalscore integer, primary
key (id)) engine=InnoDB
create table users_achievements (users_id integer not null, achievements_id
integer not null, primary key (users_id, achievements_id)) engine=InnoDB
create table users_friends (user_id integer not null, friends_id integer not
null, primary key (user_id, friends_id)) engine=InnoDB
alter table users add constraint UK_6dotkott2kjsp8vw4d0m25fb7 unique (email)
alter table activities add constraint FKq6cjukylkgxdjkm9npk9va2f2 foreign key
(user_id) references users (id)
alter table users_achievements add constraint FKg4hfidvrrl1tiwec7kfkpebf
foreign key (achievements_id) references achievements (id)
alter table users_achievements add constraint FKcfc941xlfg26nkymvu3pikrox
foreign key (users_id) references users (id)
alter table users_friends add constraint FKo51ktjiheso8mkdd5n4pdf9f3 foreign
key (friends_id) references users (id)
alter table users_friends add constraint FKry5pun2eg852sbl2l50p236bo foreign
key (user_id) references users (id)
CREATE TRIGGER RemoveActivityScore AFTER DELETE ON activities FOR EACH ROW
UPDATE users SET totalscore = totalscore - OLD.activity_amount * (SELECT
co2_savings from activitytypes WHERE id = OLD.activity_type) WHERE users.id =
OLD.user_id
CREATE TRIGGER AddActivityScore AFTER INSERT ON activities FOR EACH ROW
UPDATE users SET totalscore = totalscore + NEW.activity_amount * (SELECT
co2_savings from activitytypes WHERE id = NEW.activity_type) WHERE users.id =
NEW.user_id

```

The definition of the events that are running daily on the database

event for increasing the users score by the right amount daily

```
CREATE DEFINER=`phpmyadmin`@`%` EVENT `solarEventScore` ON SCHEDULE EVERY 1
DAY STARTS '2019-04-03 23:59:00' ON COMPLETION PRESERVE ENABLE DO UPDATE
users SET totalscore =
totalScore + (SELECT co2_savings FROM activitytypes where id = 5)
WHERE solar_panel = 1
```

event for incrementing the solar activity amount by 1 daily

```
CREATE DEFINER=`phpmyadmin`@`%` EVENT `solarEventAmount` ON SCHEDULE EVERY 1
DAY STARTS '2019-04-07 23:59:00' ON COMPLETION PRESERVE ENABLE DO UPDATE
activities SET activity_amount = activity_amount + 1 WHERE user_id IN (SELECT
u1.id from (SELECT * FROM users) as u1 WHERE solar_panel = 1)
AND activity_type = 5
```