Sao Chi Pham, Florian Murzov-Pirkl, Johanna Spaude

Lab 3: Sequence Diagrams

Lab 2 - Review

We received the use cases from Leonie Koch and Luc Nkok and read everything first. Then we discussed the use cases and scenarios and looked if anything is not clear for us.

The first point that confused us was the registration/login – process. Therefore they wrote three scenarios. In the scenario of the registration one precondition was that the user is not registered. Nevertheless, one step in the scenario checked if the user was already registered. The double check was unneeded but it was a good idea to split the three processes into three scenarios.

One big scenario was the process of ordering a hoagie. They described each step of building a hoagie, even with the maximum of each ingredient. This was nice because we got a good idea of this process. But is the order of the ingredients important for making the hoagie? In the end, they just wrote that a user can go through all steps of building a hoagie again if she/he wants to order more hoagies. This is too much work for the user though. The user could instead just state the number of hoagies the user wants. In this way the user just has to build each hoagie once.
The payment in this scenario was also very specific. There were three steps, but each one with an if-statement. Instead of this a more general overview for the payment would have been clearer.

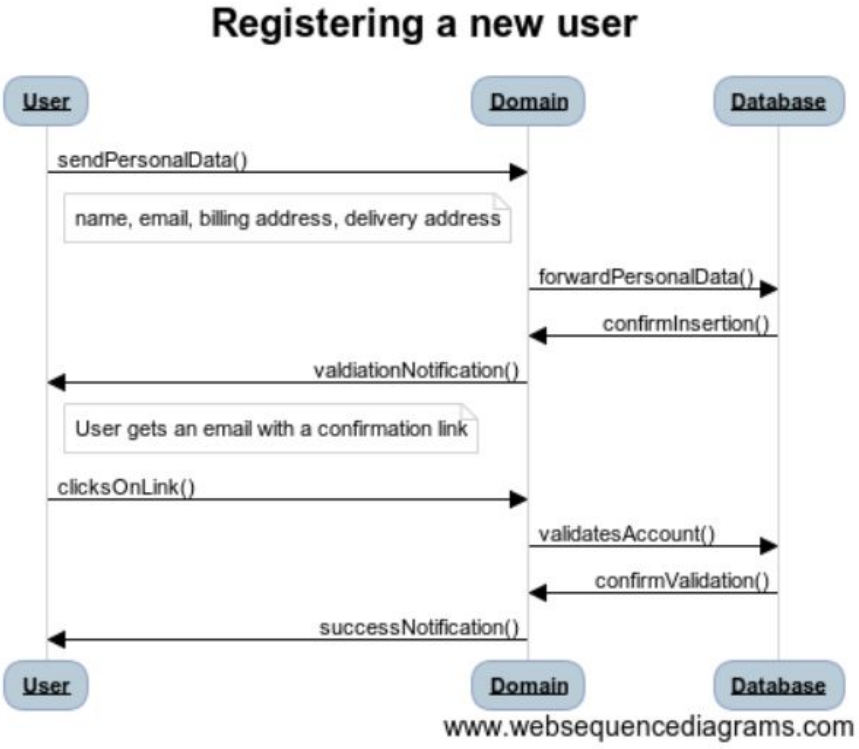
Another point we were not sure about was the packing of the food. The hoagies got packed from the food preparer and from the deliverer as well in two different scenarios. Is it important to pack the hoagies twice if the customer does not come to pick up her/his hoagie at the store? We came to the conclusion that the Hoagie is not packed twice but is being packed for transportation (put into a bag or inside a car).

We also had some questions: What happens to the money and the hoagies when the hoagies cannot be delivered? Does a customer gets five euros every time he posts a selfie?
Is the statistic report only available each month? Is only a CSV export possible? What can be done with the statistics? Most were solved in our diagrams by extending the scenarios or are just a matter of preference like the data type of the statistics. Only the question about unsuccessful orders stays but that is only due to the lack of details in the scenario we received.
Also, we discussed the names of the actors. Many times the administrator acted like a database or a domain. The names were a little bit confusing in some places.

Next, we thought about where we would input the cupcake ordering. This was easy. We would just put it into the ordering process. It could be a step after building the hoagie (point 6). The customer could also give the number of cupcakes he wants. We also thought about cool cupcake creations. But this was useless in the end, because we did not need the cupcakes anymore.

Sequence Diagrams

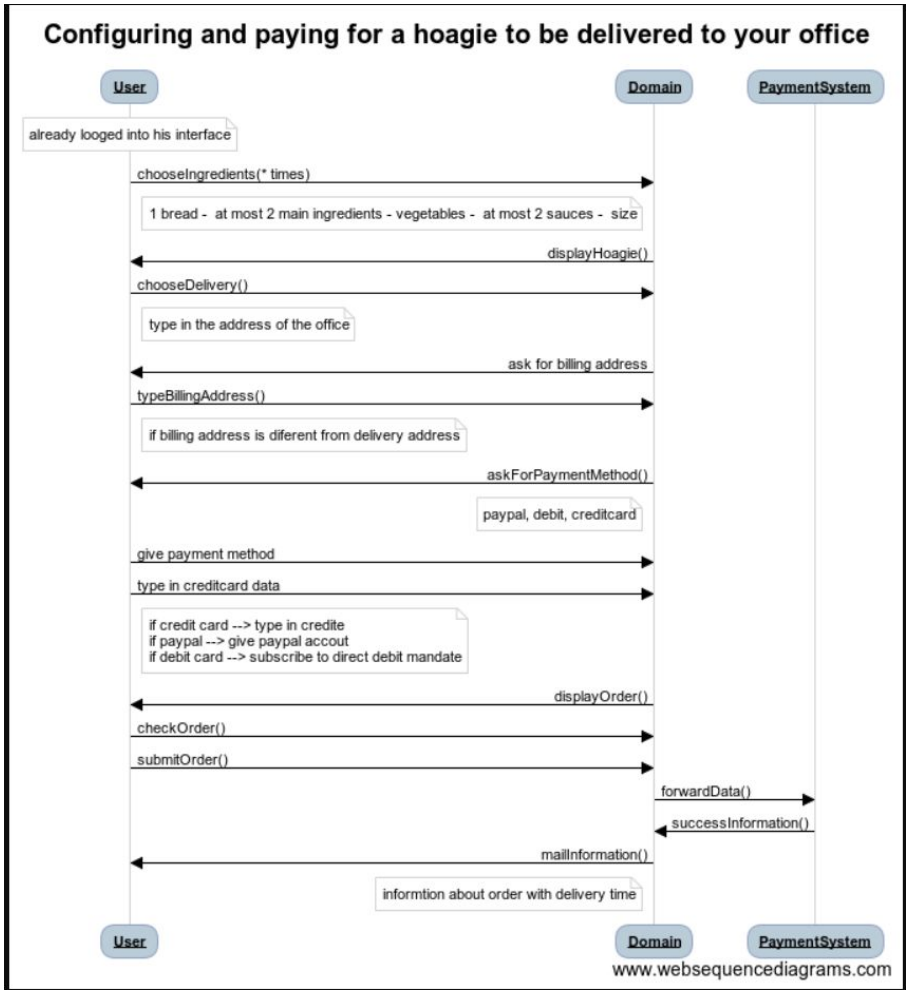
We needed pretty long for these steps. At the end of the first lab, we started to create our first sequence diagram with the given web-tool. We decided to do the first of the given use cases and worked with the scenario that Leonie and Luc had written for “Registering a new user”. We changed the names of the actors several times and had to think about the order of the steps and which step is connected with which other step. We decided to generalize and connect some steps of the login/registration. We also wrote the steps as methods for a better reuse afterwards. It was really exhausting and we needed the remaining time of the first lab for that first sequence diagram.



In the second lab, we roughly reviewed our work from the week before and decided to work on separate sequence diagrams and compare afterwards.

Johanna:
I took the use case of ordering a hoagie. Like we did it the week before I just wrote the actors first and then the steps as connection between the actors. An easy change was that the delivery address just has to be the address of the office. But after a short time I got confused because the process of paying the hoagies was really big and had also may actors. There were the PayPal website, the debit mandate and the domain of the hoagie store involved, but not all actors every time. After discussing with Florian and Chi I decided to put this whole payment process into an easier sequence of steps and connected the

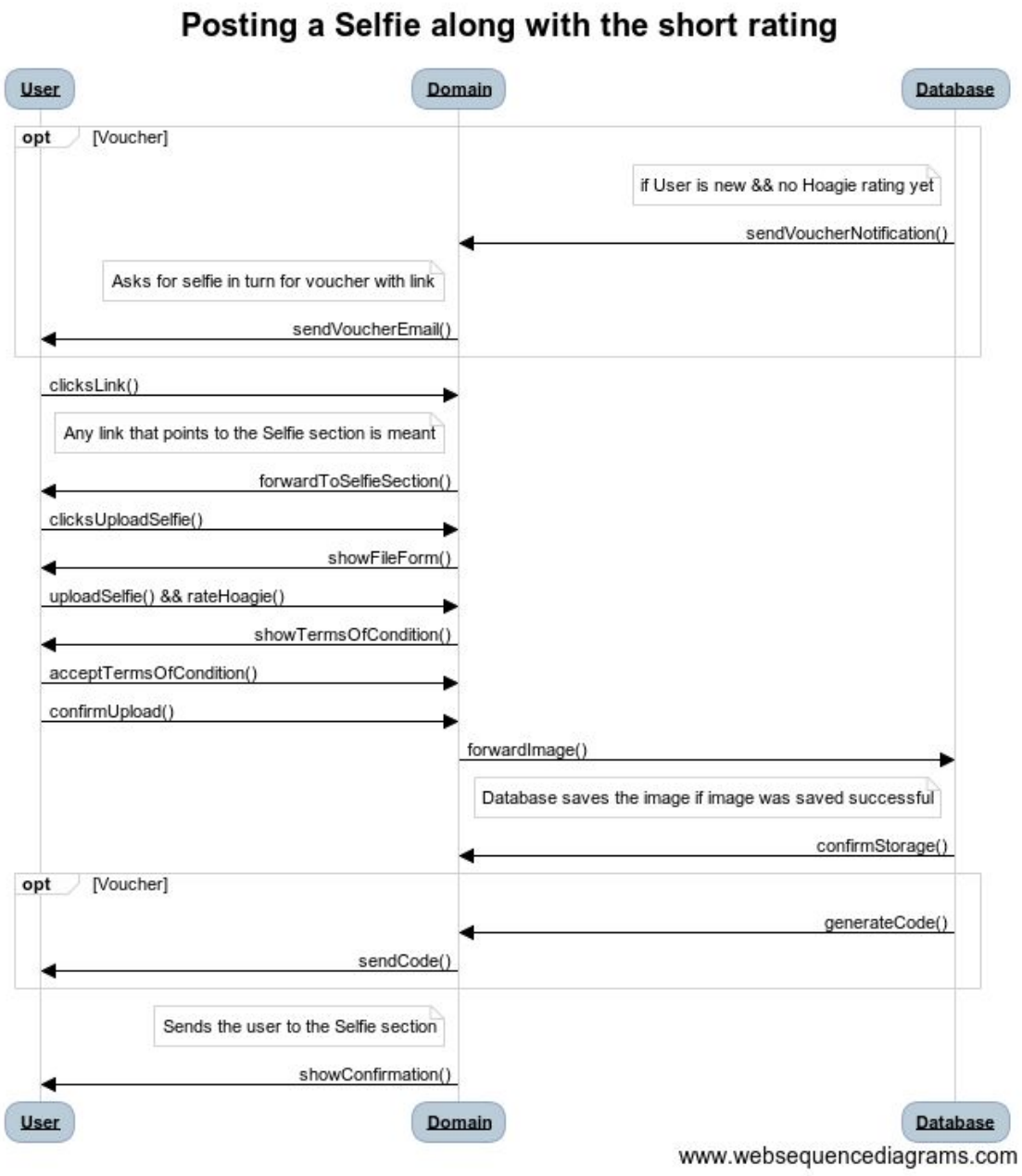
things related to the payment into the Payment System. I also decided that the user has to give his data to the Domain in all cases and that the domain forwards this data to the specific payment system. This looked much better in the end.
Another point that confused me was that the user had to pay first before checking and confirming his order. I would never pay before checking if I am ordering the right things would I? So I changed the order of the steps so that the user can check his order first. In the end the Sequence Diagram of ordering hoagies to an office looked like this:



Sao Chi:
My case was posting a selfie with a short rating. Luckily they had a scenario called Selfie which I could reuse step by step. We called the website or the user interface with which the customer interacts Domain. Additionally, we had also the actors User and Database.

One thing that troubled us at first was that there was no restriction on how many vouchers can be send to a customer in the description from Luc and Leonie. To prevent the shop from becoming bankrupt after a huge Hoagie order, I limited it to one voucher for the first time a user gets a Hoagie delivered.
After a while, I noticed that the voucher was only an optional part of the whole process described in Leonie's and Luc's report. It is much nicer if customers could post as many selfies and ratings as possible and not only on the first order because this way the shop

looks more popular. So, I marked the sections optional in the diagram that were only related to it.



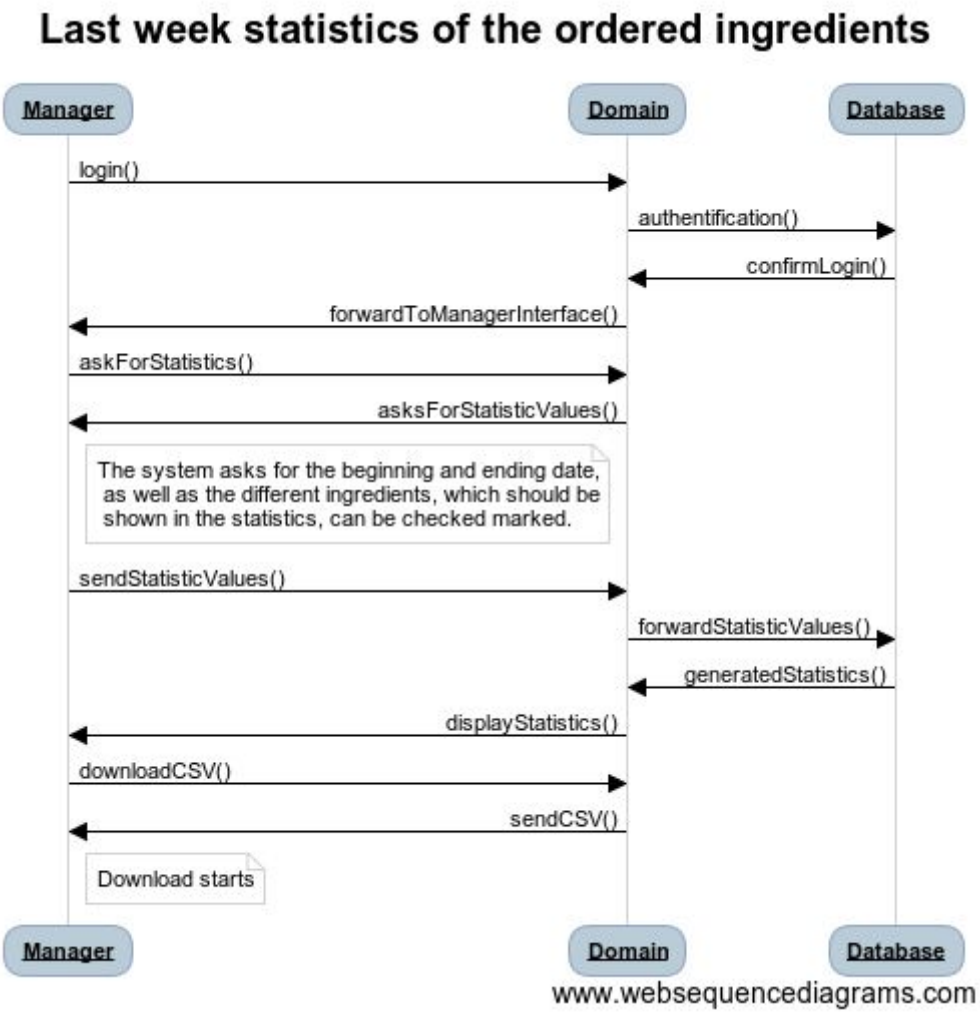
Another concern was privacy. To ensure that data is not collected in secret, the images are only saved on success and only after the terms of condition/privacy note has been accepted by the user. One decision we made while doing the sequence diagrams was that the database would only communicate with the domain and that the user could only communicate with the domain. This way malicious input cannot be sent directly to the database or data from the database cannot be sent accidentally, and some checks could be done beforehand.

Florian:

I draw a sequence diagram for the task “Obtaining information on how much of each ingredient has been ordered for the past full week”. The use case I was working with (from Leonie and Luc), was called “Monthly statistics”, which had the same idea behind, but for the

whole month instead of the last week, only. As in our first sequence diagram, I split the actor “Administrator” into two other ones, “Domain” and “Database”.

Name	Monthly statistics
Actors	administrator, manager
Precondition	The system is running at least for one month and orders have been made in order to get statistics
Sequence of steps	1. manager logs in to backend 2. manager clicks on “generate monthly report” 3. the report shows Income and expenditure, amount of consumed ingredients, new and existing customers, and sales per day 4. he/she can export the file as csv
Postcondition	In the end the manager has a montly report



The manager has probably its own user interface on the domain, where she/he logs in. That is most likely meant by the first point. In the sequence diagram, I described it as the Manager “login()” on the Domain. Then the Domain has to ask for an “authentification()” by the Database. If the login-data was correct, the Database “confirmLogin()” to the Domain, which then “forwardToManagerInterface()”. This part was the first sequence of steps in the use case of Leonie and Luc.

The second step is, that the manager clicks on a button to generate the statistics. Since the use case of our sequence diagram is about making a statistic of each ingredient of the past full week, I tried to make both use cases (the one of our task and the one from Leonie and Luc) possible. First, the Manager clicks on a certain button on the Domain in order to “askForStatistics()”. Then the Domain “asksForStatisticValues()”, such as an beginning or ending date and all the ingredients she/he wants to find in it. In that manner, it is possible to specify, if the statistic should just be about the full last week (e. g. Monday-Sunday) or the last month or just any other particular time the manager wishes to have a statistic about. Same with the ingredients, it should be possible to choose just a single one, particular ones or all of them by check marking them (for instance). Once the Manager has filled it out, she/he “sendStatisticValues()” to the Domain, which “forwardStatisticValues()” to the Database.
In case the Manager chooses dates, when no statistics were collected (e. g. before the shop even started to operate or dates in the future), we decided it will change the dates to the first or last possible dates and show a message about it to the Manager. We forgot to add a note about it in the sequence diagram.

Step three is basically when the Database generates the statistics. It probably makes sense, if it counts all received orders on the chosen period by default. Maybe when filling out the statistic values, there can be even added more options (if the Manger wishes), about the beginning and ending time on the specified dates and if only orders, which were successfully paid and delivered counts into it and so on. The “generatedStatistics()” are sent to the Domain, which “displayStatistics()” to the Manager.

In step four, the Manager decides to “downloadCSV()”, because it is the file type she/he wants to have in the use case of Leonie and Luc. Since the statistics are already available at the Domain, it directly “sendCSV()” to the Manager, which starts the download of the file.

Evaluation

Johanna:
I was really surprised how much time we needed for just reading and discussing the scenarios and use cases from Leonie and Luc. They wrote all things very detailed what was okay for me. I had a good idea of there hoagie store but there were also some unneeded steps. I discovered the mistake in the ordering process when I wrote the sequence diagram. Before that I did not see that a user has to pay his order before he can check it. I really like the idea of the sequence diagrams. The use cases were a little bit to abstract for me as they just show which actor is interacting with other things. But I can imagine the sequences really good. I mostly worked with the scenarios and not with the use cases during the labs.
Time:
I needed both labs and one and a half hour additionally for understanding the processes and writing the report.

Flo:

The sequence diagrams were confusing to me at first, but after modelling the first one, it suddenly felt way easier. I especially noticed, they give me a much better idea how I would implement it later, than by just having the use cases. And in a different way, it shows what details must be considered. For instance, it probably makes sense in many cases to just call one actor “administrator” in the use cases, but in the sequence diagrams, it was confusing to us. Splitting it into two, then we suddenly had to care about an additional actor, but somehow it was more logical to work in that way for me. I really like the sequence diagrams also for understanding a process someone else has modelled. When we compared the work of each other, we quickly could get the ideas and thoughts and make suggestions to them.
Time: In addition to the two labs, I needed another 1:15 hour to write my part of the report.

Sao Chi:
I enjoyed making sequence diagrams more as they show what and where something happens in the process. They also show pretty early where logic errors are and what pieces are missing. The sequence diagrams that we made reminded of the client-server model since the use cases and scenarios assume that everything is done online.
Time: The time in the labs were used for the reading, discussing the use cases and scenarios, and making the sequence diagrams. Additionally, I needed one and a half hour to write the report and correct my diagram.

Appendix

Report with scenarios and use cases from Leonie Koch and Luc Nkok.

Luc Nkok, Leonie Koch

Lab 02: Scenarios, Use Cases, and Use Cases Diagrams

In order to develop use case scenarios, we went step by step through an ordering process on basis of our experiences with web applications. Beginning with the registration and ending with drawing statistics we tried to develop standard and exceptional cases and to sum them up in a use case diagram. Here are our use cases:

Name	User registration
Actors	User, administrator
Precondition	User is not registered, email address is not known to the system
Sequence of steps	1. User calls domain 2. Already registered? Go on with “Login” 3. Input personal data 4. confirmation mail
Postcondition	User is logged in and data is saved in database

Name	Forgot Password
Actors	User, administrator
Precondition	User forgot his password, email is already known to system
Sequence of steps	User clicks on “forgot password! User needs to type in his email User receives a “reset password” mail User confirms by clicking on link User types in new password
Postcondition	User data is updated, user can login

Name	Login
Actors	User, administrator
Precondition	User is already registered (if not, go on with “User registration) and knows his password (if not, go on with “Forgot password”)
Sequence of steps	User types in email and password and is able to order things
Postcondition	User sees the interface for logged in users

Name	Order process
Actors	User, administrator
Precondition	User wants to order something and is at home/ at work, user is logged in and sees the interface for logged in users
Sequence of steps	1. Choose 1 bread 2. Choose at most 2 main ingredients 3. Choose vegetables (default is everything, user can ‘uncheck’ items) 4. Choose at most 2 sauces 5. Choose between regular and large size 6. If user wants another hoagie, repeat step 1 to 5, elsewhise go on with step 7 7. Choose pickup or delivery 8. Choose delivery address

	9. If delivery address is not equal to billingsadress, user also needs to give billing address details 10. Give payment details (paypal, debit, creditcard) 11. If paypal forwarding to paypal website 12. If creditcard type in creditcard data 13. If debit card subscribe to direct debit mandate 14. Control order and submit 15. confirmation mail with estimated pick up/ delivery time
Postcondition	Order is delivered to foodpreparer

Name	Food preparer get information to prepare hoagie(s)
Actors	Food preparer, administrator, customer
Precondition	Customer order has been received, Food preparer is registered and can be assigned to the system, administrator sends information to food preparer about composition of the hoagies
Sequence of steps	1. Food preparer takes the form sends by the adminisrator 2. Food preparer starts to prepare food using the form giving by the administrator 3. After the preparation food preparer starts to pack the food 4. Food preparer confirm that food is ready for collection and delivery
Postcondition	Administrator get information that the procduct is ready for delivery food preparer confirm that he is available to prepare an next order

Name	Product delivery
Actors	Deliverer, administrator, customer
Precondition	Administrator received the information that the food can be delivered Deliverer is registered and known by the administrator (system) Deliverer confirm that he is free and available to deliver
Sequence of steps	1. Deliverer gets information to deliver the product 2. Deliverer collects and packs the product 3. Deliverer drives to the address of the customer 4. Deliver hands over the product to the customer 5. Deliverer confirms delivery
Postcondition	Hoagies has been delivered to the customer The order is complete and Deliverer is ready for a new one

Name	undeliverable
Actors	Deliverer, administrator, customer
Precondition	Order was submitted and delivery attempt made
Sequence of steps	1. customer was not neither present nor reachable 2.Supplier indicates why order can not be delivered and cancels the order 3. customer receives information mail 4. After unsuccessful delivery, the order will be returned
Postcondition	Order was canceled and customer informed about reason

Name	Selfie

Actors	Deliverer, administrator, customer
Precondition	Delivery has been successfully delivered and customer is currently eating
Sequence of steps	1. Supplier confirms delivery 2. Customer receives email "We hope you enjoy it, get your 5 Euro voucher now and upload a photo of yourself with Hoagie" 3. Customer clicks link in email 4. Customer comes on rubric Selfies and Comments 5. Customer clicks on "Upload photo and save voucher" 6. Upload file form - Customer uploads image from own files 7. Customer adds short comment of eating experience 8. Customer must accept Terms and Conditions and Privacy Notice 9. Customer clicks "upload image" 10. Image is stored in database to which manager has access → statistics 11. Coupon code will be generated and sent by email to customer 12. Confirmation screen "Thank you for attending, you will receive mail from us"
Postcondition	Customer can view his picture and that of many other happy customers in the Selfie and Comments section and share them through many social networks

Name	Monthly statistics
Actors	administrator, manager
Precondition	The system is running at least for one month and orders have been made in order to get statistics
Sequence of steps	1. manager logs in to backend 2. manager clicks on "generate monthly report" 3. the report shows Income and expenditure, amount of consumed ingredients, new and existing customers, and sales per day 4. he/she can export the file as csv
Postcondition	In the end the manager has a montly report

As ingredients, we set the following items:

Bread
White bread
Full grain
rye

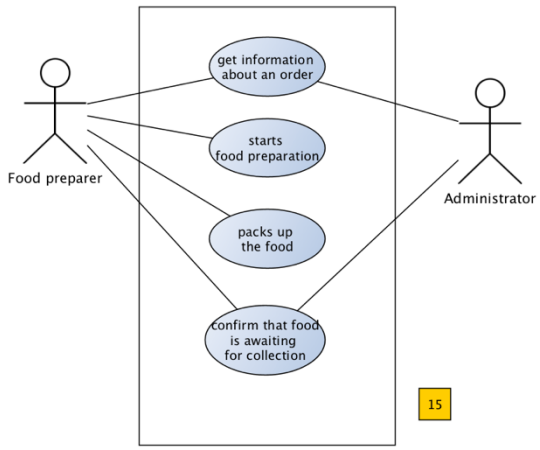
Main ingredients
Turkey
beef
chicken
cheese
tuna
pulled pork
tofu
Vegetables
salad
aragula
onions

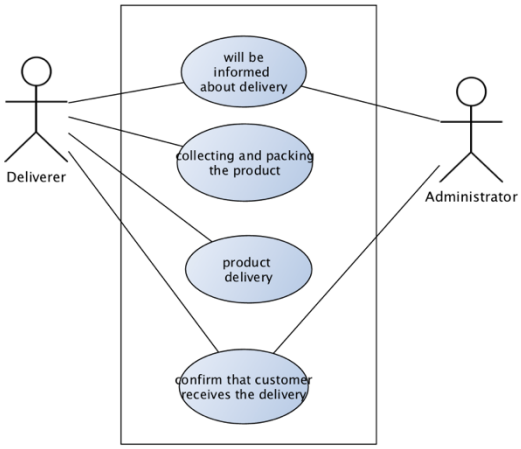
tomatos
cucumber
jalapenos
corn
Sauces
garlic
herbs
curry
hot
peanuts
ketchup
mayo

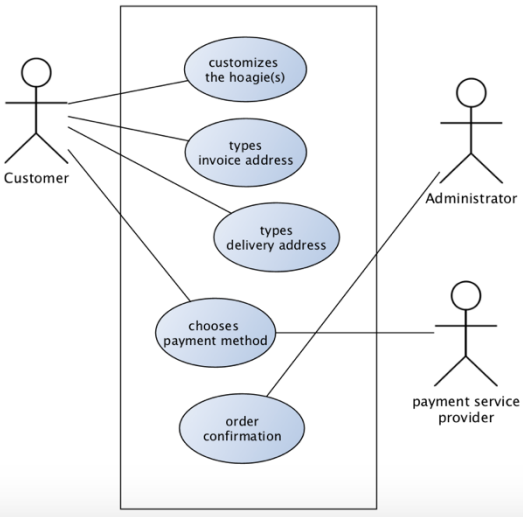
The actors in our scenarios are:

User: the customer, who uses the application to order hoagies
Administrator: the admin controls the technical part of the ordering process, cares about the system
Food preparer: prepares the food
Payment service provider: services to pay the food, as for example paypal, maestro, etc.
Manager: he wants to gain information and statistic about the user behavior
Deliverer: picks up the prepared food and brings it the customer

We started to make a use case diagram for each use case:







Luc Nkok, Leonie Koch

To combine all those use cases in a general use case diagram we took all our use cases and all affected actors and drew the following diagram:

Time: 7 hours
Reflection:

Luc: At the beginning of this exercise we started to create a lot of scenarios. Each of us developed at least 4-5 scenarios. Until we realized that most of the scenarios resembled each other or did not refer to our main business process, like what has the system to do and not how it has to do. As a result, we tried to find most important requirements of our system and to reflect them as use-case diagrams.

Leonie: We made the experience that it is very easy to run into too much detailed scenarios. At first, it does not matter what exactly an automated confirmation mail looks like or how exactly food compilations might look like. We learned to not get too far into detailed processes but to look at the overall context at first, which actors are to be considered and which scenarios. This is much more difficult because which seems unimportant for now, might be a “project killer” later.