Frankfurt University of Applied Sciences
Faculty 2 Computer Science and Engineering
Computer Science (B.Sc.)

Project Home Delivery

Software Engineering Analysis

Submitted by:

Ceylan Yildirim, 1211500
Lamiaa Kamar, 1336304
Thanh Tung Trinh, 1320718
Nouhaila Ben Messaou, 1336333

Professor: **Alexander Mützel**

Date 05.03.2021

Table of Contents

Project Home Delivery ***1***

Introduction ***3***

Declaration of Authorship **3**

Software Description **4**

Definition of product backlog **4**

System Architecture ***5***

Requirement Analysis ***6***

Backlog ***12***

Functional Requirements **12**

Non-functional Requirements **13**

Project schedule ***14***

Protocols of the meetings ***15***

Sprint planning #1 ..... **15**
Use case description 16

Sprint review #1 ..... **20**
Item reviews performed by Ceylan Yildirim 20
Item reviews performed by Tung Trinh 21
Item review by Kamar Lamiaa 21
Item review by Nouhaila Ben Messaou 21

Results of sprint #1 ..... **22**
RAM1 - Registration and management 22
RAM2 : Creating Profile 29
RAM3 : Settings 30
PDO1 – Placement of delivery orders 32
POD2 41
DA - Delete Account 49
RT – Rating System 56

Sprint planning #2 ..... **62**
Use Case Description 62

Sprint review #2 ..... **64**
Item reviews performed by Ceylan Yildirim: 64
Item reviews performed by Tung Trinh: 64

Results of sprint #2 ..... **64**
RAM4: Changing password 64
OPC1 – Order Pickup by Contractor 69
OPC2 : Contractor's perspective 73

Sprint planning #3 ..... **78**

Sprint Review #3 ..... **82**
Item reviews performed by Lamia Kamar: 82
Item reviews performed by Nouhaila Ben Messaou: 82

Results of sprint #3 ..... **82**
FACP - FAQ and Contact us platform 82
RS- Reward System: 88

Introduction

Declaration of Authorship

We hereby certify that the project report we are submitting is entirely our own original work except where otherwise indicated. We are aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Ceylan Yildirim

Nouhaila Ben Messaou

Thanh Tung Trinh

Lamiaa Kamar

Software Description

As a start-up company, we received the request to plan a home delivery software. Because of the pandemic crisis a lot of shops, pharmacies, and restaurants needed to newly introduce home delivery services very quickly. At the same time, unfortunately many persons have lost their jobs in bars, shops, etc. Our idea is to bring these two groups of people together by offering a home delivery platform. Everybody who needs a delivery service can register and ask for a delivery. At the same time, contractors can also register and pick-up delivery orders from the system.

Definition of product backlog

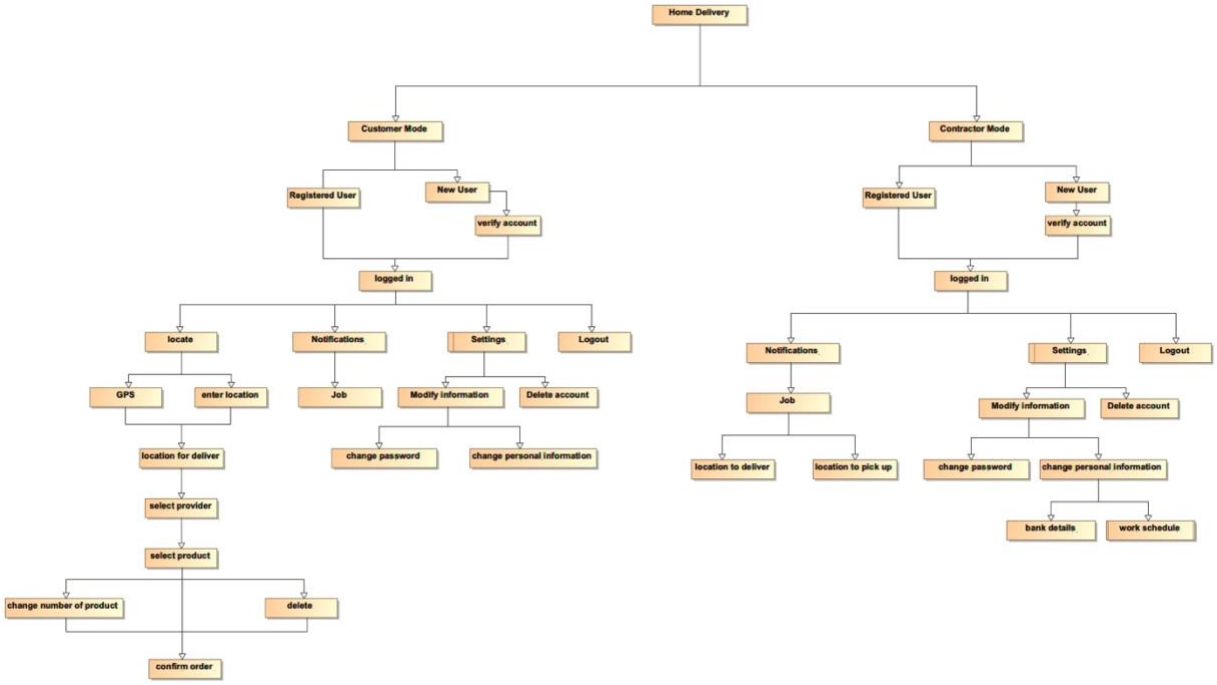
A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.

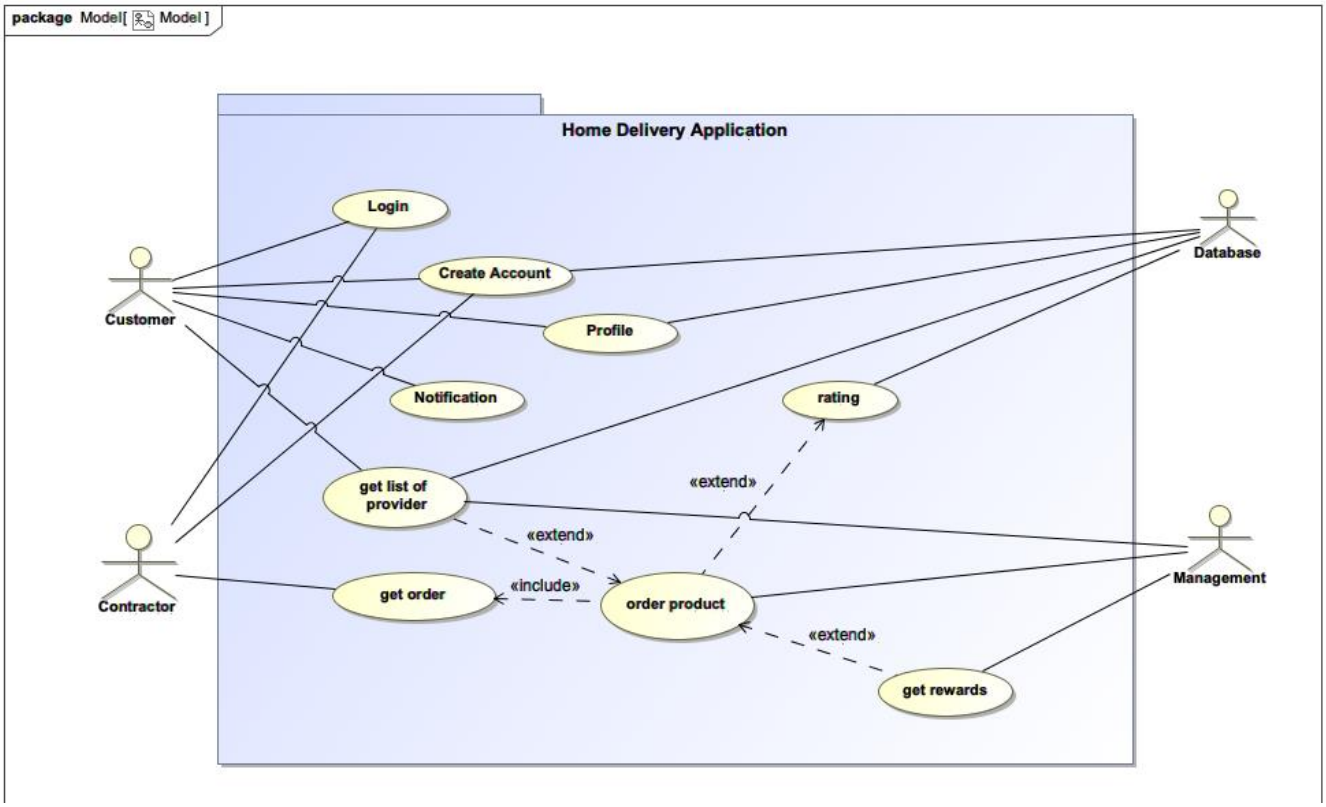
Nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.

It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome.

Product backlog items take a variety of formats, with user stories being the most common.

System Architecture





Requirement Analysis

Home-delivery Analysis Project					
Theme/ Subject area	ID	User Story Description	Definition of done	Business value	Status /Sprint
Customer registration and management	**RAM1**	new customer/ contractor: Creates an account by choosing his role (customer / contractor) to get access to our App. Creates new account by signing up, and agreeing with the Terms of Service and Privacy Policy and entering personal information (first name, last name, email, password).	The customer/contractor can create an account successfully and use our application flawlessly.	High	DONE/1
		Signs up without creating an account for ease of use for people who want to use our application only once/who want to test our application	The customer/contractor use our App without creating an account	Medium	
	RAM2	Customer / Contractor: Creates profile by Adding personal information (address, phone number, payment methods) to the	The customer/contractor should create a profile	High	DONE/1

		profile to start with the application Changing password	after the registration to simplify the process of the delivery The customer/contractor is able to change his password for safety reasons		
	RAM3	Customer / Contractor: Can use Settings, help and notifications. Can login / logout	The customer/contractor is able to profit from our services The customer/contractor can login and logout without any problem	High	DONE/1
	RAM4	Customer / Contractor: When the user forgets his password, he will get a new password request per e-mail to change his password by clicking on the link in the email.	In case the customer has forgotten his password, he can be able to change his password successfully	Medium	DONE/2

Placement of delivery orders	**PDO1**	Customer: Should accept that the App receives his current location via GPS in order to get list of providers near him and get his order. (delivery ONLY in a radius of 5km). Alternative: enters the desired delivery location manually.	The user accepts to share his location with the App and gets his order. In case the order is for someone else, the location can be given manually.	High	DONE/1
	PDO2	Customer: Shop from different providers in the radius of 5 km	The customer can select different products from different providers and Add them to cart without a problem and receives his order as expected	High	DONE/2

Order pick up by contractor	**OPC1**	Admin: Decides who receives which order by using an algorithm which allows him to choose the suitable contractor and its mission. The contractor will be selected after these priorities: Distance, activity status and rating reviews	With the help of the Algorithm, the right contractor is selected to do his mission successfully.	Low	Done/1
	OPC2	Contractor: Gets informed by the App about his new mission, and goes to the provider to pick up the order and deliver it to the customer.	The order is delivered to the customer punctually.	High	Done/2
Delete Account	**DA**	Customer /Contractor: When the user deletes his account, it will be deactivated for 30 days if he logs in during this time his account will be reactivated if not all his data has to be deleted after 30 days. The data required for economic analytics has to be saved in a way that the user is untraceable.	The customer is able to delete his account and be sure his data is deleted and won't be misused	Medium	DONE/1

FAQ and Contact us platform	**FACP**	Contractor: wants to get competent assistance if he faced problems and can be able to reach customer service for questions that he can't find answer to or leave comments.	The customer service is organized. The FAQ webpage is organized and there shall be employees assisting and answering customers questions asked via contact us form.	Medium	Done/3
Rating System	**RT**	Customer: After the contractor confirms the delivery the customer gets a notification to rate the service using stars' classification system, which he can either accept or skip. The customer is asked to rate the following aspects: -Friendliness -Time -State of the order	The customer is able to rate our App, or skip the rating.	Low	Done/1

Reward System	**GRS**	Customer: Orders from the App to earn points After this he can get different rewards (e.g. free delivery after 7 orders, discounts)	after ordering from the App, the customer receives his points and have the choice to use them or keep collecting them to get bigger discounts.	Low	Done/3
Social media and Advertising	**SAA**	Company: Increases the Number of the customer by advertising under the condition that the revenue should be higher than the costs of advertising.	Reaching people through our Ads, and being known	Low	Done/3

Backlog

Functional Requirements

- Selecting between customer or contractor
- Creating account
- Validation account
- Changing password
- System provides a database
- Log in and log out is possible for the user
- User gets a list of shops
- User gets notifications
- Multiple payment options
- Resetting or changing password
- Display 2 options of entering location
- Scanning location through GPS
- Display all available providers
- Display all available products
- Saving selected products into shopping cart
- Change/remove items in shopping cart
- Data (time and costs of the trip) must be saved in our database
- Algorithm to sort and select contractors
- Display Order Information
- Display Provider Location
- Display Customer Location
- Display Delivery Time
- the program allows deleting the data and user accounts.
- User is able to rate our App

Non-functional Requirements

- Enough capacity of database
- Protection against overload of the database
- Payment method shall guarantee modern encryption standards and has to be valid
- Security (encryption password)
- Security (resetting password via email)
- Availability of employees should be granted
- User has to enter his location
- User has to confirm his order
- filtering the data, in which the program analyst selects what should be deleted and what not.
- A working webserver and a database
- Service should be available from any web-browser or by installing apps.
- The user interface should be as simple possible.

Project schedule

This table presents an overview about the steps and the status of this project

Task	Start	End	Duration	Status
Meeting 1	18.12.2020	23.12.2020		
Definition of 'Done'	18.12.2020	19.12.2020	2 days	Done
Engineer requirements	18.12.2020	21.12.2020	3 days	Done
Develop product backlog	18.12.2020	22.12.2020	4 days	Done
Estimate backlog items	22.12.2020	23.12.2020	2 days	Done
Meeting 2				Done
Sprint planning #1	24.12.2020	24.12.2020	1 day	Done
Meeting 3	10.01.2021	16.01.2021		
Sprint review #1	10.01.2021	10.01.2021	1 day	Done
Retrospective	11.01.2021	14.01.2021	3 days	Done
Sprint planning #2	16.01.2021	16.01.2021	1 day	Done
Meeting 4	21.01.2021	23.01.2021		
Sprint review #2	21.01.2021	21.01.2021	1 day	Done
Retrospective	22.01.2021	23.01.2021	2 days	Done
Sprint planning #3	23.01.2021	23.01.2021	1 day	Done
Meeting 5	29.01.2020	30.01.2020		
Sprint review #3	29.01.2021	29.01.2021	1 day	Done
Retrospective	30.01.2021	30.01.2021	1 day	Done
Meeting 6	31.01.2021	31.01.2020	1 day	Done
Preparation of the presentation				
Presentation of the project	05.02.2021			
Submission of the written report	05.03.2021			

Protocols of the meetings

In this excerpt, the meetings (sprint planning, retrospective, and sprint review) are documented.

Sprint planning #1

The role of the product owner will be taken by Ceylan Yildirim in the first sprint. We used *Planning Poker* to estimate the backlog items together and updated the values in the product backlog.

ID	Assigned to	Begin	Finished	Workload in each item in percent
RAM1	Ceylan Yildirim	24.12.19	14.01.2021	100%
RAM2	Ceylan Yildirim	24.12.19	14.01.2021	100%
RAM3	Ceylan Yildirim	24.12.19	14.01.2021	100%
POD1	Tung Trinh	24.12.19	14.01.2021	100%
POD2	Tung Trinh	24.12.19	14.01.2021	100%
DA	Lamiaa Kamar	24.12.19	14.01.2021	100%
RT	Nouhaila ben Messaou	24.12.19	14.01.2021	100%

Use case description

Use Case field	**Our customer wants to use our Home-Delivery service. He needs to register first. If user is already registered, he just needs to login. Our management system saves the data. This use case artefact shall be developed.**
Use Case name	Customer / Contractor registration and management1 /RAM1
Subject area	Contractor registration and management
Trigger	User: • Enter password • Change password • Create account • Enter payment option • Click on settings • Click on notifications
Actors	User
Use case overview	We need personal information for creating a new account for a new customer. After confirmation the account will be ready. Already registered customer can login in several ways.
Preconditions	User accepts terms of conditions.
Termination outcome	A positive ending: User successful registered. Now he can use the application. A negative ending: User isn't successfully registered. He can't use the application without registration.
Use Case description	User can login and manage his account.
Use Case association	User
Input summary	Personal information, password
Output summary	Registration, login, settings, help, order product, check delivery
Use case notes	None

Use case description 1.1: *Use case description for RAM1, RAM2, RAM3*

Use Case Field	**Our User wants to order products from different providers nearby. He is able to select and add items to cart or remove from cart.**
Use Case Name	Placement of Delivery Order
Subject Area	Placement of Delivery Order
Trigger	User: • Click on enter location • Click on scan location • Click on provider • Click on add to cart • Click on remove from cart • Click on confirm order
Actors	User
Use case overview	An overview of all nearby Providers with all Products.
Preconditions	User enters his location
Termination outcome	A positive ending: User is able to look for nearby providers. User has a large overview over all items and providers. User can choose different products from different providers. A negative ending: User can't scan his location. GPS connection is failed. User can't have the overview of Providers and Items.
Use Case description	He logs in to order items from nearby shops. Order will be delivered after the confirmation has been passed.
Use Case association	User, Database
Input summary	The user, the selected provider/shop, the selected items
Output summary	Different items from different providers

Use case notes	Order information, arrived time has to be displayed in the end

Use case description 1.2: *Use case description for POD1 and POD2*

Use case field	**The user wants to delete his account.**
Use case name	Deleting account
Subject area	Customer registration and management
Trigger	Delete account option in the settings menu.
Actors	User, Program, Analyst
Use case overview	If the users wishes to delete his account, then we should make sure all his data is deleted and his needed data are kept in a way so that he can't be tracked.
Preconditions	A registered account.
Termination outcome	Positive ending: User successfully deletes his account. Negative ending: User fails to delete his account.
Condition affecting termination outcome	If the user can't delete his account due to an error in the program. That means there are bugs in it.
Use case description	User no longer wishes to use our services and wants to delete his account. He can delete it from app or webpage. The account isn't deleted immediately, at first it's deactivated for 30days, if the user doesn't log in during this time his account will then be permanently deleted.
Use case associations	Registration , login
Input summary	Make sure the user wants to delete his account.
Output summary	Ensuring that all user data gets deleted
Usability index	5

Use case description 1.3: *Use case description for DA*

Use case field	the ability to rate the App, after a successful submission of the merchandise.
Use case name	Rating System
Subject area	Customer satisfaction
Trigger	Notification after receiving the order
Actors	The Customer
Use case overview	After receiving the order, the customer is able to rate the App so we can improve our customer service
Preconditions	The question has to be related to our delivery services.
Termination outcome	Positive ending: our customer is satisfied and gives more than 3 stars to our services Negative ending: our customer rates less than 3 stars or skips the rate which make this functionality pointless
Condition affecting termination outcome	If our customer didn't get the notification to rate our App, because of a problem in our server
Use case description	The customer receives his order and rates our App depends on the following services: punctuality and friendliness of the contractor, and the state of the order, and can also skip the rate as well.
Use case associations	Notification
Input summary	Rate 1 to 5 stars
Output summary	Increasing or decreasing of our stars
Usability index	4

Use case description 1.4: *Use case description for RT*

Sprint review #1

In the following section, the first sprint review is presented. Normally, only the product owner decides whether the produced artefacts fulfill the *definition of done* or not. In order to share the work of the product owner, we decided that each team member performs the sprint review of another team member. The aim is to distribute the workload of this project and to decrease the overhead which may occur if the product owner is responsible for each produced artefact. Each member of the group should correct and enhance her/his own task before the second sprint begins.

Item reviews performed by Ceylan Yildirim

RAM1

- Separation the registration of customer and contractor
- No access without login
- Changed validation to email
- Increased capacity of customer
- Registration app changing into Registration app/web

RAM2

- Saving address of customer for faster order
- Payment methods have to be entered
- Increased options for payment
- Option to change password for safety

RAM3

- Contact support is given for problems that may arise
- Updated Notifications with news about new products
- Added profile settings (e.g. for changed phone number, address or payment methods)

Item reviews performed by Tung Trinh

POD1 and POD2:

- removing database as an entity
- database has a new role as actor
- activity diagram has decisions
- label the arrows of the activity diagram
- removing the payment from the use case diagram

OPC1 and OPC2

- estimate the effort required to implement the requirement
- sequence diagram is hard to implement

Item review by Kamar Lamiaa

DA:

- Remove entering login data again to delete account
- Business related data isn't deleted
- Deactivate account for 30 days before deleting it

Item review by Nouhaila Ben Messaou

RT:

- Notification after the order
- Add to: time, friendliness and state of order a rating option to know what exactly can be improved
- Option to skip the rate

Results of sprint #1

RAM1 - Registration and management

As new User / new Contractor	Users have to select between customer and contractor.
Users have to create account for using home delivery application	User can create new account. By signing up, the user agrees with the Terms of Service and Privacy Policy. For creating an account, the user has to enter personal information (first name, last name, email, password).
User can sign up without creating account	User can use the application by signing up via Facebook or Google.
Account validation via mail	Users have to validate account via mail to confirm account.

For UI related functions: UI prototype

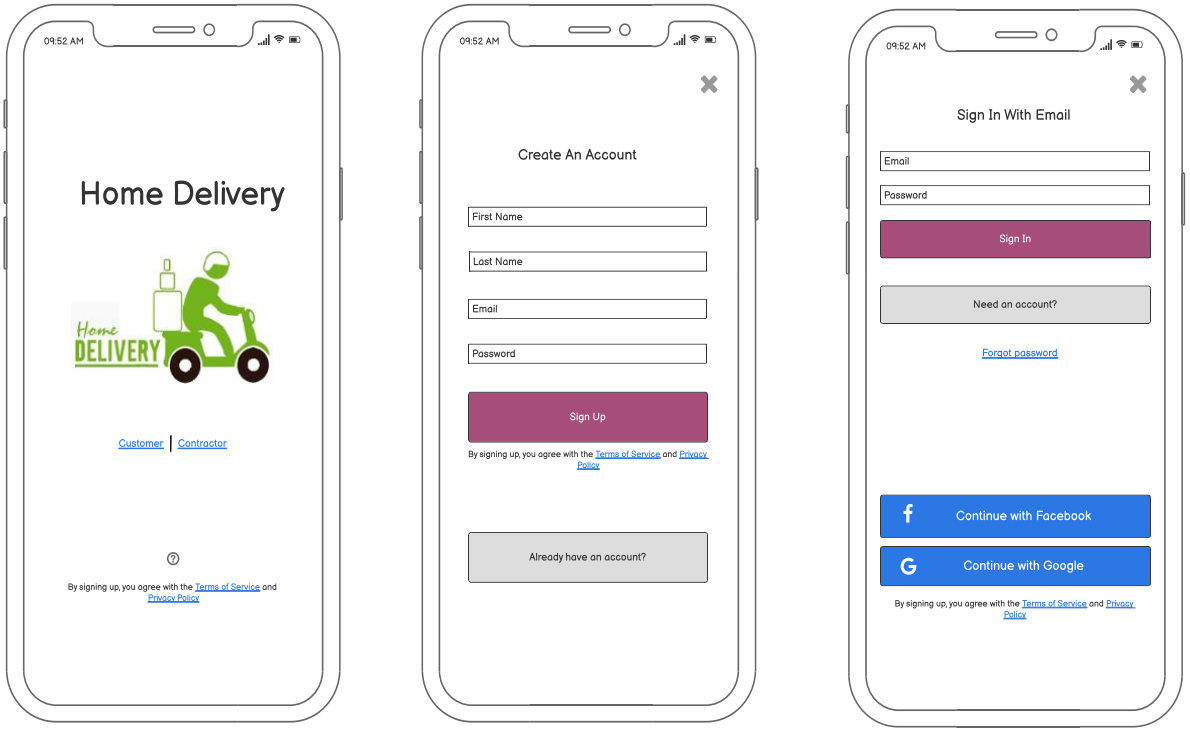


Figure 1.1: *Mobile GUI prototype login*

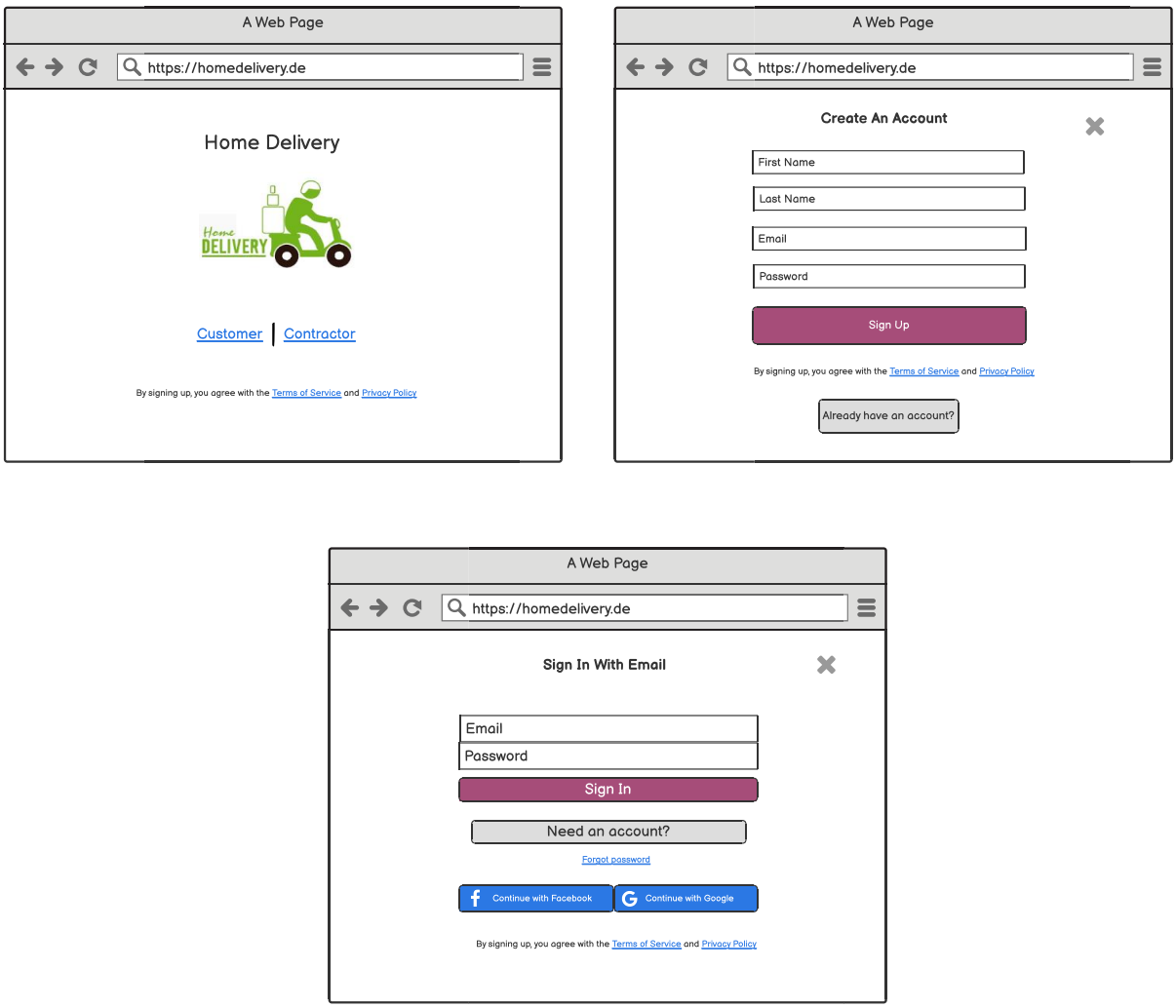


Figure 1.2: *WEB GUI prototype login*

Business value of the corresponding functionality

It is important that the system works properly and fulfils customer requirements so that more and more customers will register. If you have many customers, it increases the business value.

Effort estimation for the implementation of the requirement

Estimated, the implementation can take four to five days and the development of a database- entity-relationship model can take up to three days.

In contrast, experienced programmers need 1 to 2 days to implement and up to 1 day to develop a database property relationship model.

Another example of effort estimation depends on how many customers are registered and if something does not work, the problem has to be solved, so it takes roughly 1 to 2 days to fix the errors.

UML Diagrams:

Use Case Diagram

The user can select between customer or contractor. Both of them have the same options. They first can register if they don't have an account yet or login. If the password is wrong, they can change password. The user also has the option to login

via Facebook or Google.

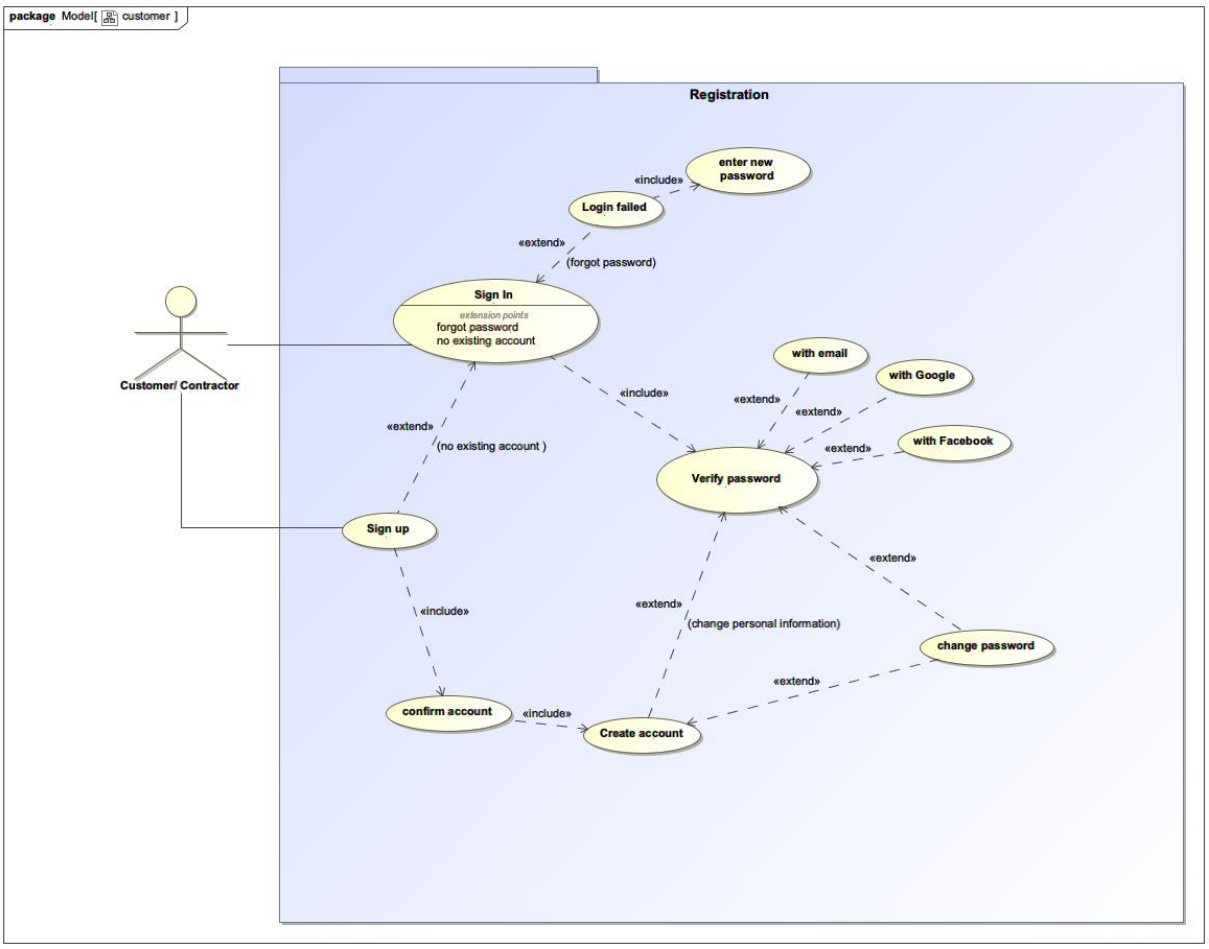


Figure 1.3: Use Case Diagram

Actor Customer / Contractor

- User performs Sign In
- User performs Sign Out

UseCase Sign In

- Sign in is performed by Customer / Contractor
- Sign in extends Login failed

UseCase Login failed

- Login failed extends Enter new password

UseCase Sign up

- Sign up includes Confirm account

UseCase Create account

- Create account extends Verify password
- Create account extends Change password

UseCase Verify password

- Verify password is generalized by E-mail
- Verify password is generalized by Google
- Verify password is generalized by Facebook

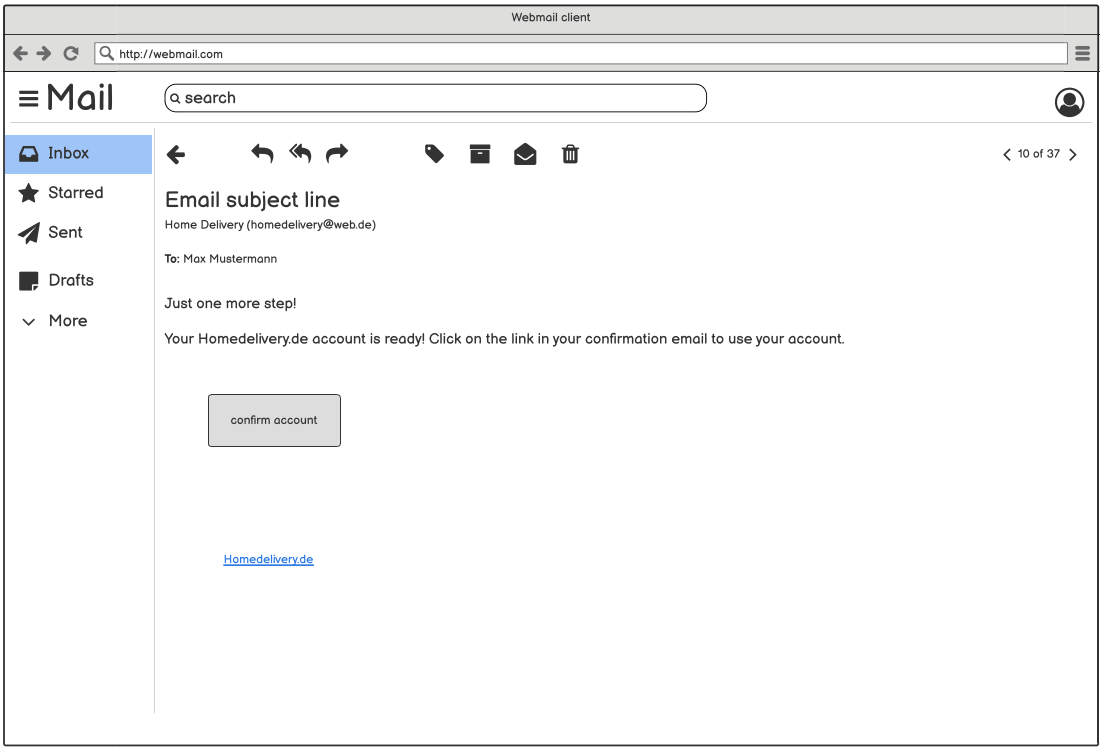


Figure 1.4: *WEB GUI prototype validation mail*

After Registration the user have to verify his account via E-Mail to make sure that it is a real person.

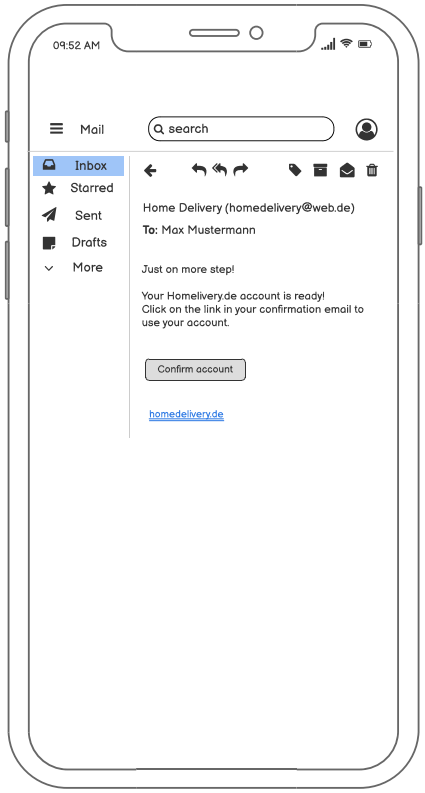


Figure 1.5: *Mobile GUI prototype validation mail*

RAM2 : Creating Profile

As User / Contractor	
Creating profile	Adding personal information (address, phone number, payment methods) to the profile to start with the application.
Changing password	User can change his password.

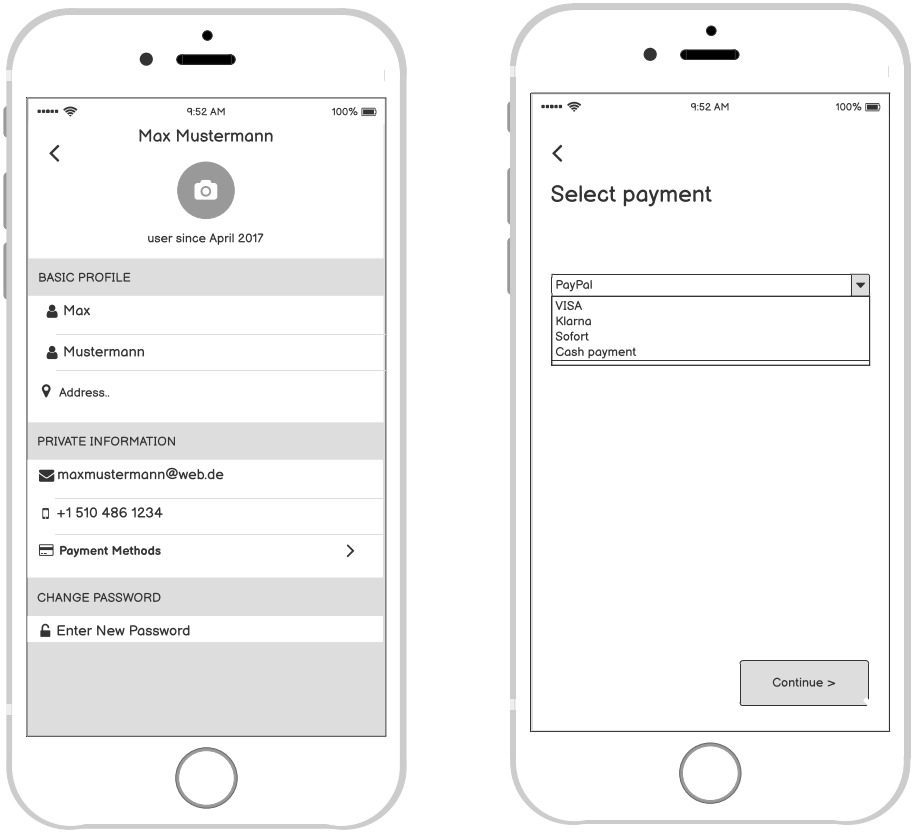


Figure 2.1: *Mobile GUI prototype profile*

RAM3 : Settings

As User / Contractor Settings Login / Logout	We should provide all needed requirements for a user account for example: settings, help, notifications.

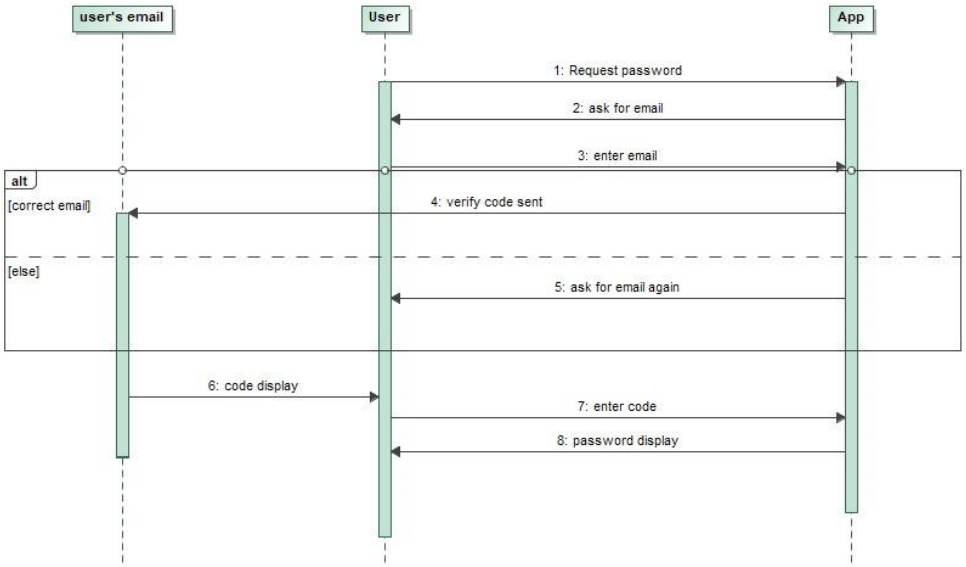


Figure 2.2: *Sequence Diagram*

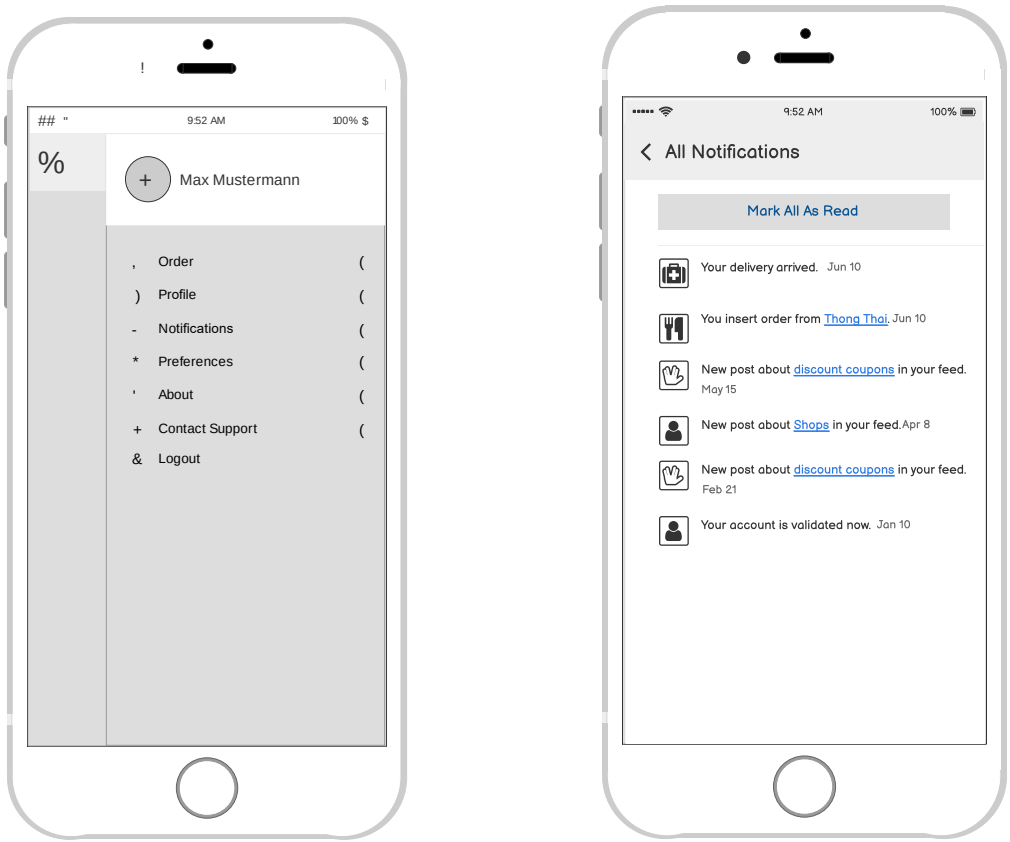


Figure 2.3: *Mobile GUI prototype Logout / Notifications*

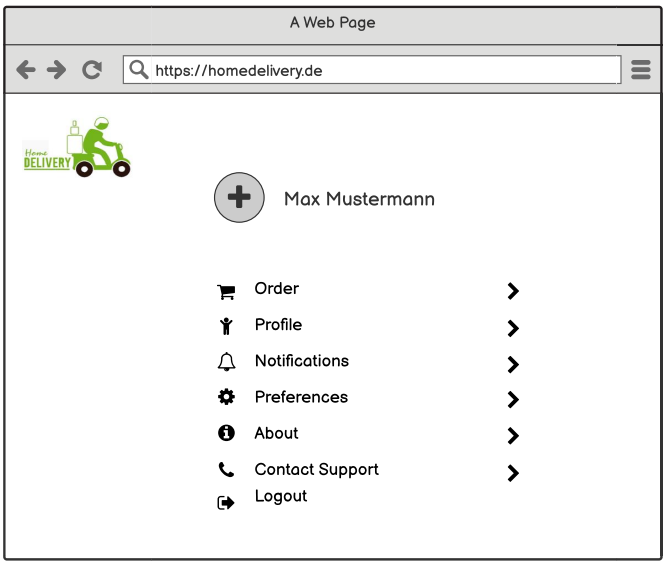


Figure 2.4: *WEB GUI prototype main menue*

PDO1 – Placement of delivery orders

As User I need to let the app scan for my current location through GPS or I enter the location where the Orders should be delivered.	We only deliver only in a radius of 5km. User has to enter the location in order to get back a list of Providers near him. User can only choose the products from the providers in the radius of his given location.

For UI related functions: UI prototype

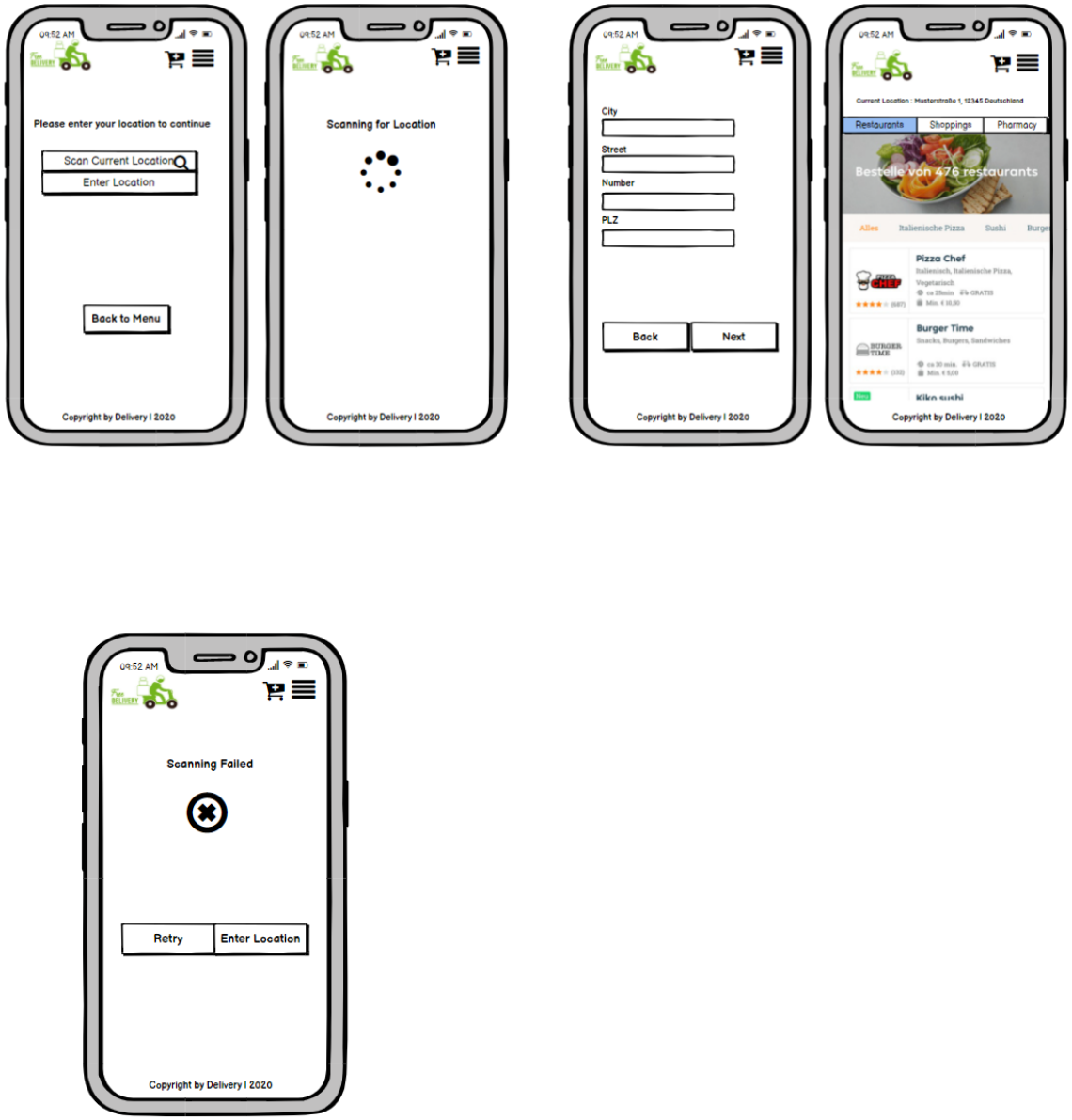


Figure 3.1: *Mobile GUI prototype validation mail*

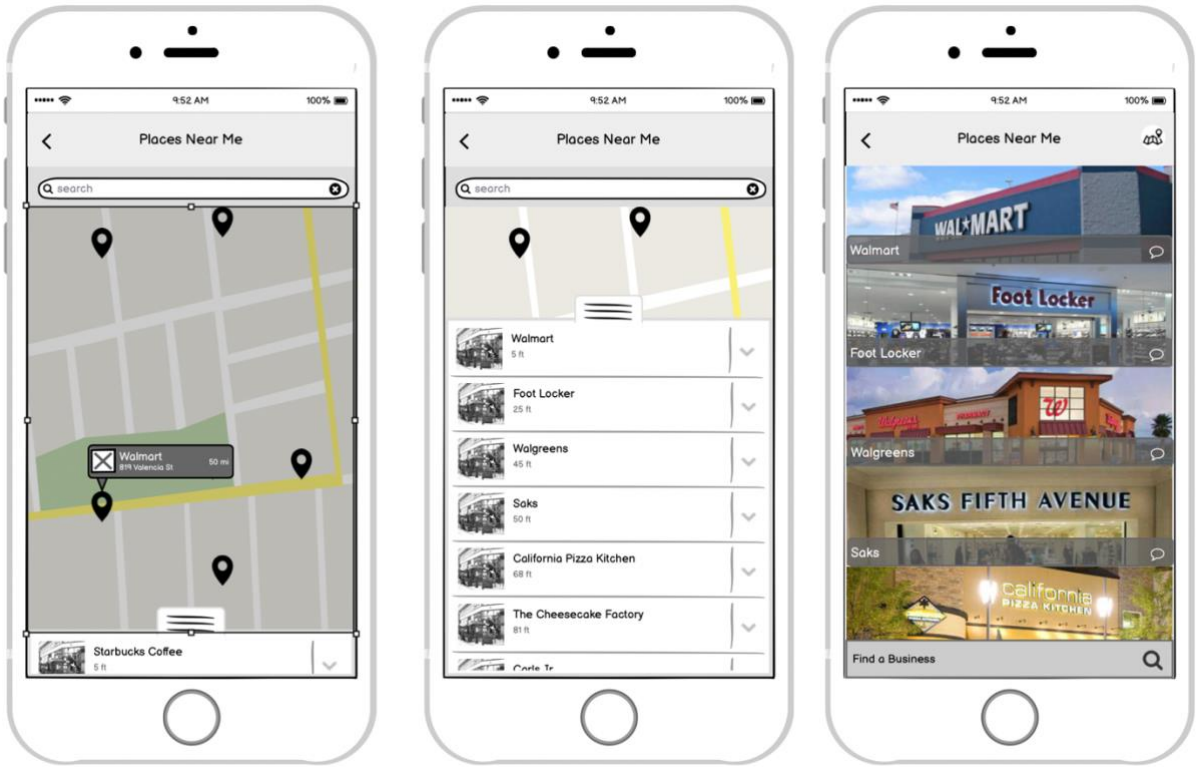


Figure 3.2: *Mobile GUI prototype nearby providers display*

Figure 3.3: *WEB GUI prototype enter location*

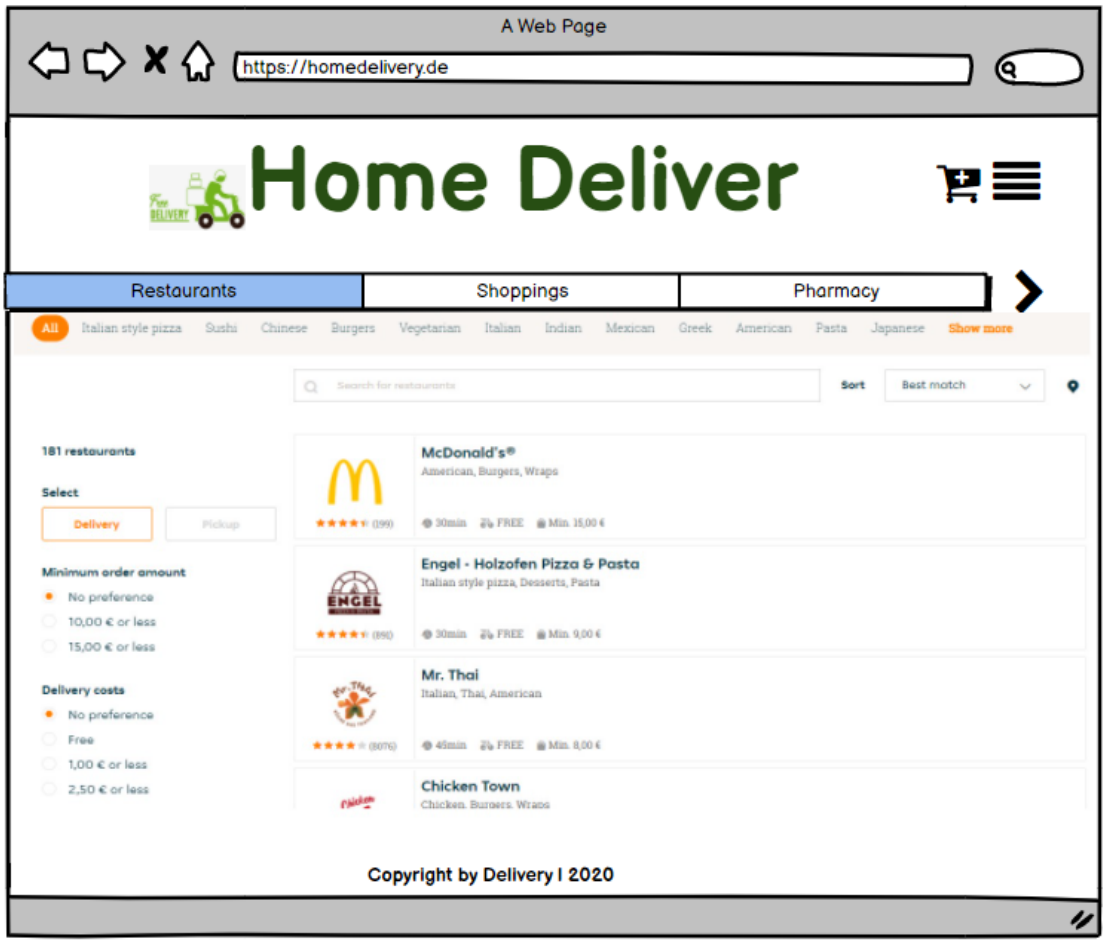


Figure 3.4: *WEB GUI prototype providers display*

Categorization of requirement(functional/non-functional)

Functional requirement:

- Display 2 options of entering location
- Scanning location through GPS
- Display all available providers

Non-functional requirement:

- User has to enter his location

Business value of the corresponding functionality

It is very important to know the current location or the entered location of the user. Because we only deliver in the radius of 5km, user will be able to order only products in this radius. We choose this radius to minimize the costs of transportation and maximize the profits. 5km radius is also fit for an average waiting time.

Effort estimation for the implementation of the requirement

We need a team of programmers. All the Providers information will be saved into the database. For this functionality, the database needs to interact with the locations of providers.
The application has to be able to analyze the location of user correctly. With experienced programmers, we can scan and analyze the location within a couple of seconds. After that, it will give the user a clear view of all available providers near him in the radius of 5km.

UML Diagrams

Use case diagram:

Our main actor is user. The user enters the location to use the service. At this point, we can see how user can get a list of providers. There is possibility that the scanning is not possible, and user has to enter his location manually.

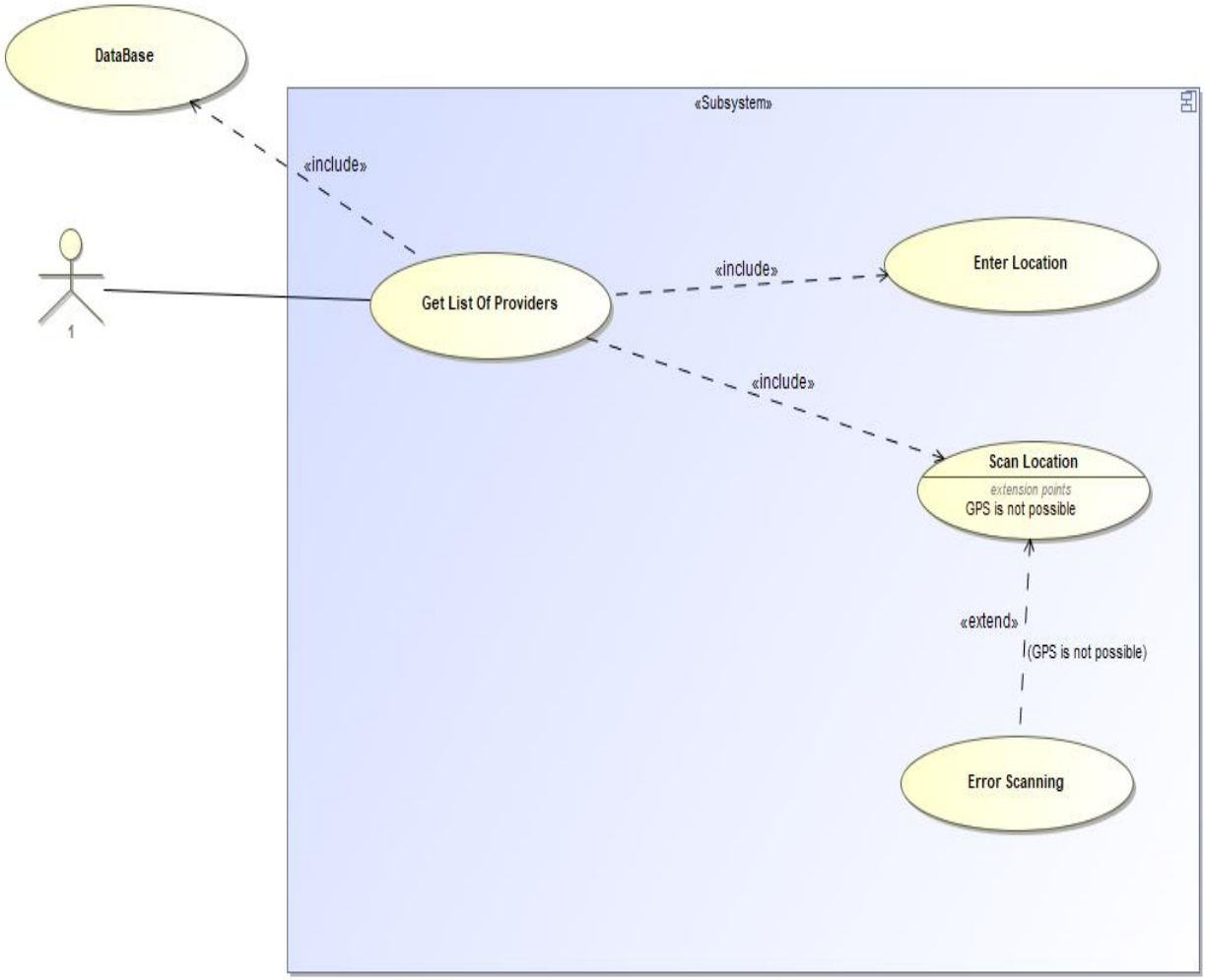


Figure 3.5: *Use Case Diagram*

Actor (User)

- User performs UserGetsListOfProviders

(Database)

- Database is included by UserGetsListOfProviders

UseCase UserGetsListOfProviders

- UserGetsListOfProviders includes UserEnterLocation
- UserGetsListOfProviders includes Database
- UserGetsListOfProviders includes UserScanLocation

UseCase UserEntersLocation

- UserEntersLocation is included by UserGetsListOfProviders

UseCase UserScansLocation

- UserScansLocation is included by UserGetsListOfProviders
- UserEntersLocation extends UsersErrorScanning
- Extension Points:
    1. GPS not allowed
    2. Bluetooth is off

UseCase Error

- Error extends UsersScansLocation

Activity diagram:

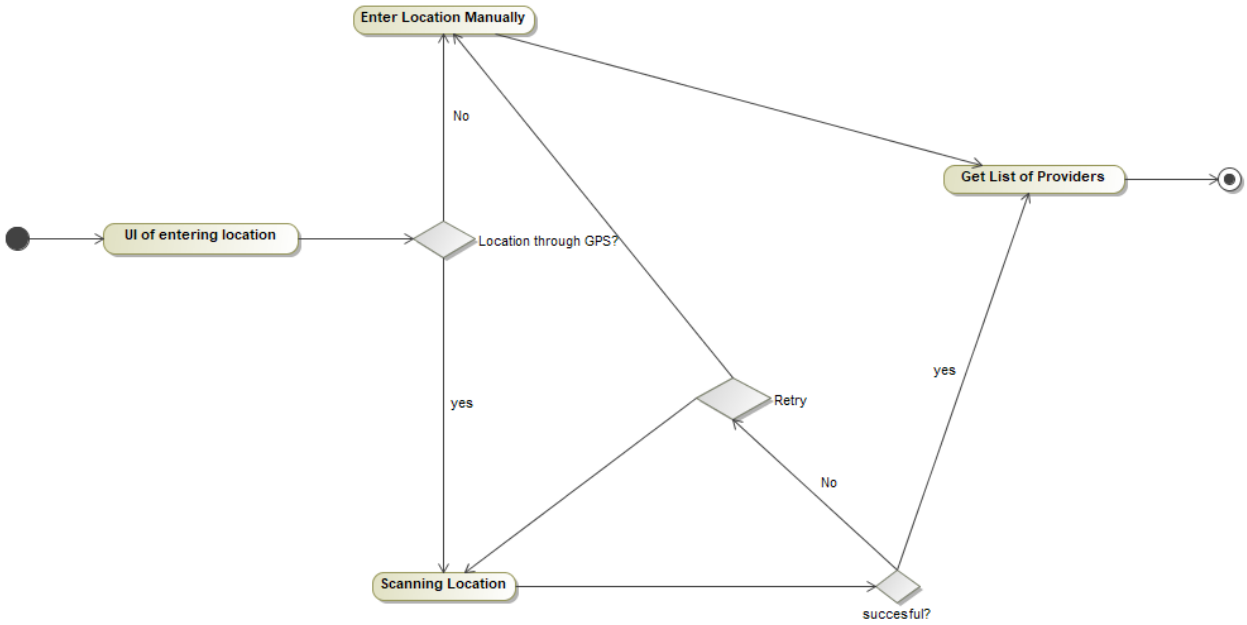


Figure 3.6: *Activity Diagram*

Sequence diagram:

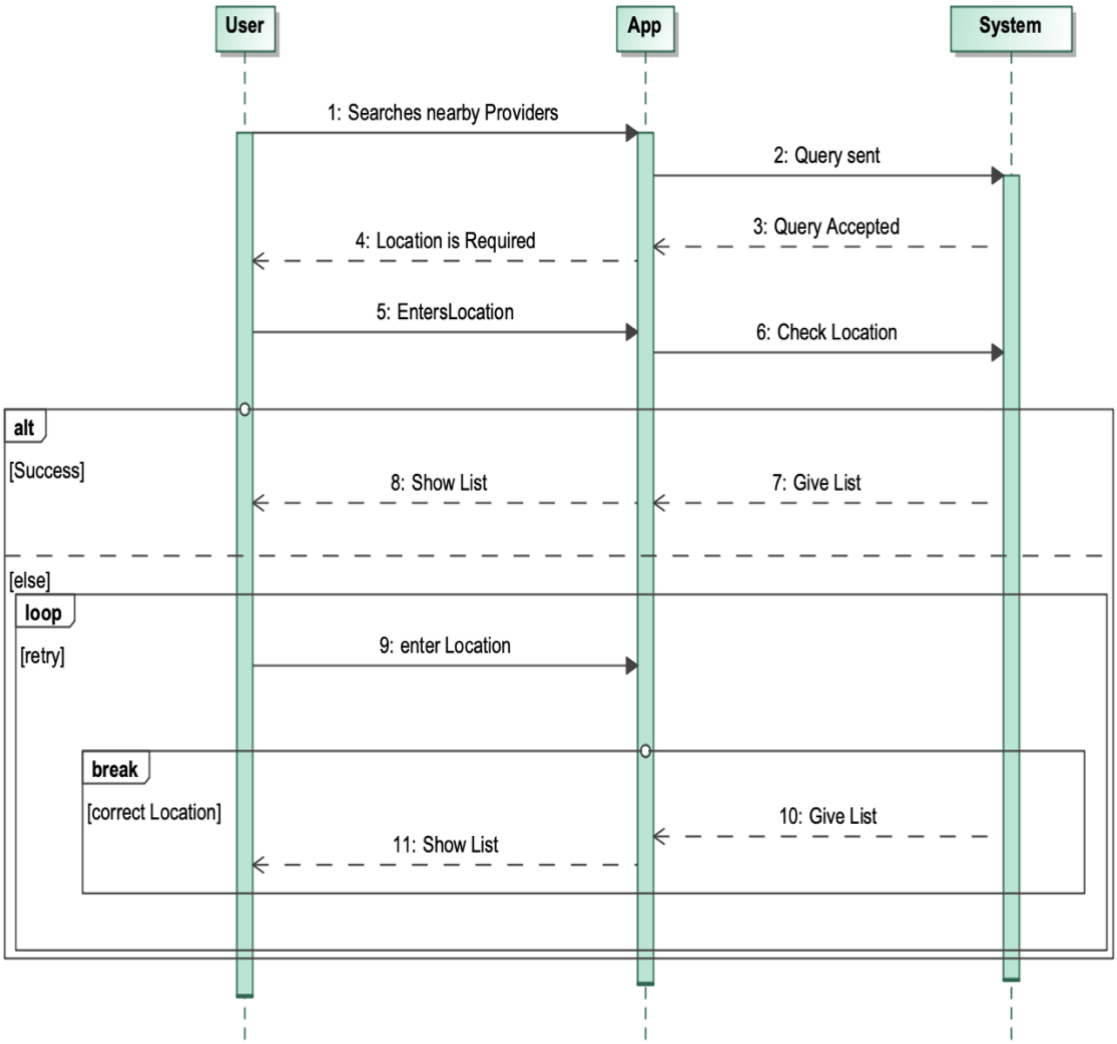


Figure 3.7: *Sequence Diagram*

POD2

As User, I can select different products from different providers and add them to cart.	The user will have a clear overview of all available providers. User can choose any products from any providers. User can change or add the number of products. User can also remove items from cart.
In the end, I can remove items from cart or change the number of items. After that, I am able to confirm my orders and wait for the order to be delivered.	In the Shopping cart, user will see the total amount of costs.

For UI related functions: UI prototype

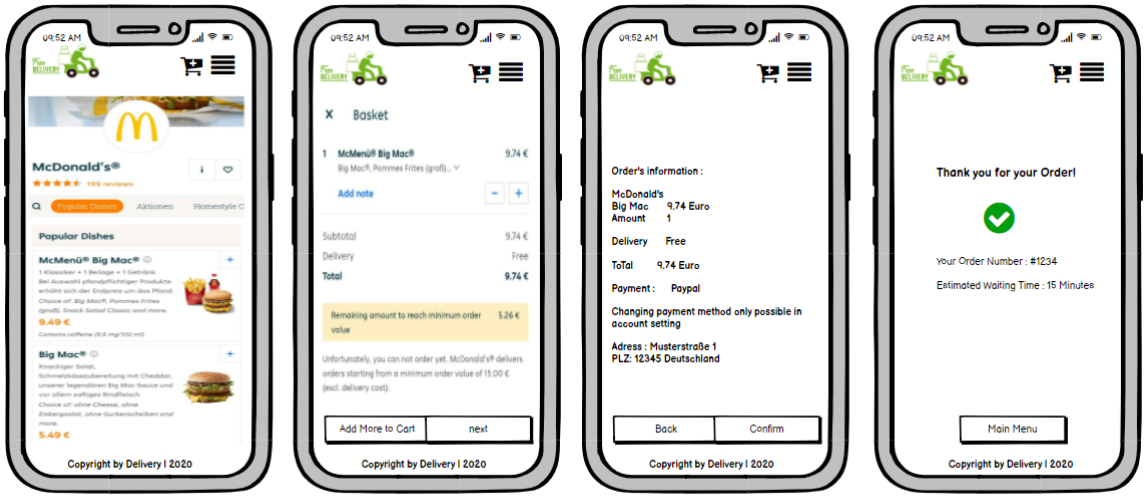


Figure 4.1: *Mobile GUI prototype order*

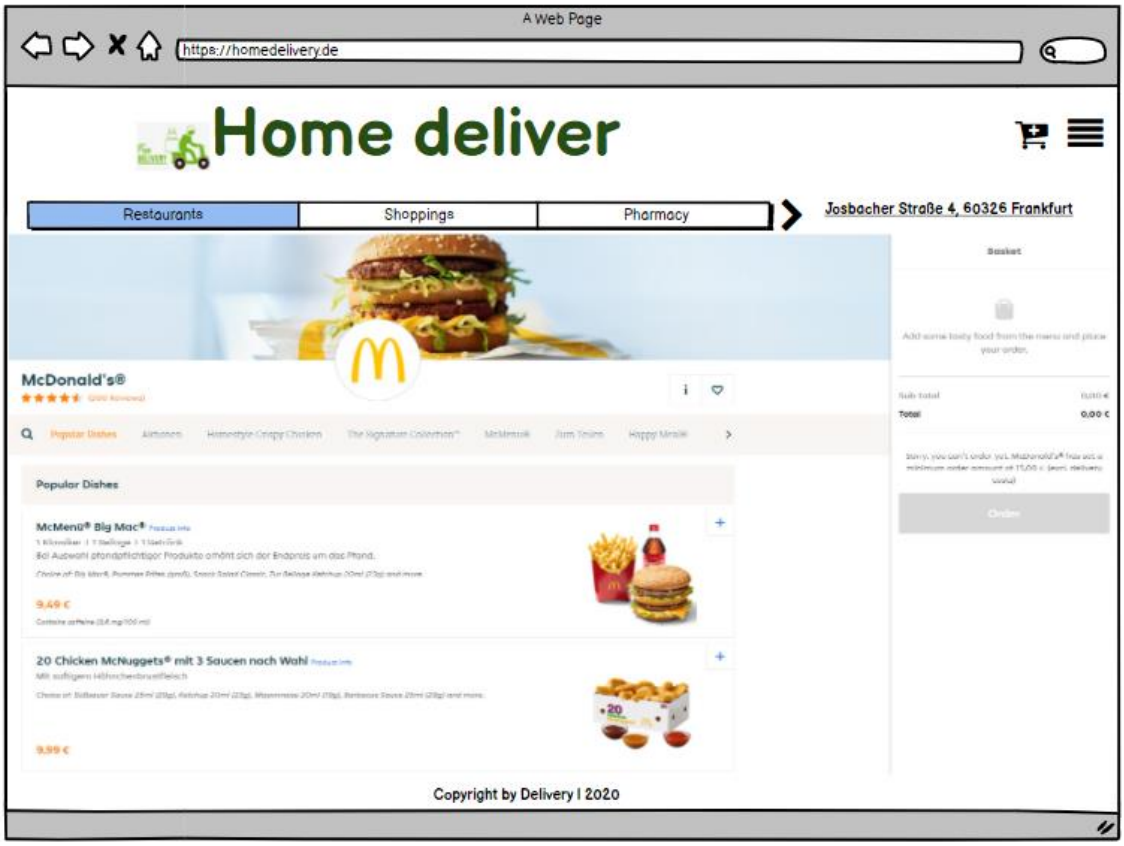


Figure 4.2: *Web GUI prototype provider*

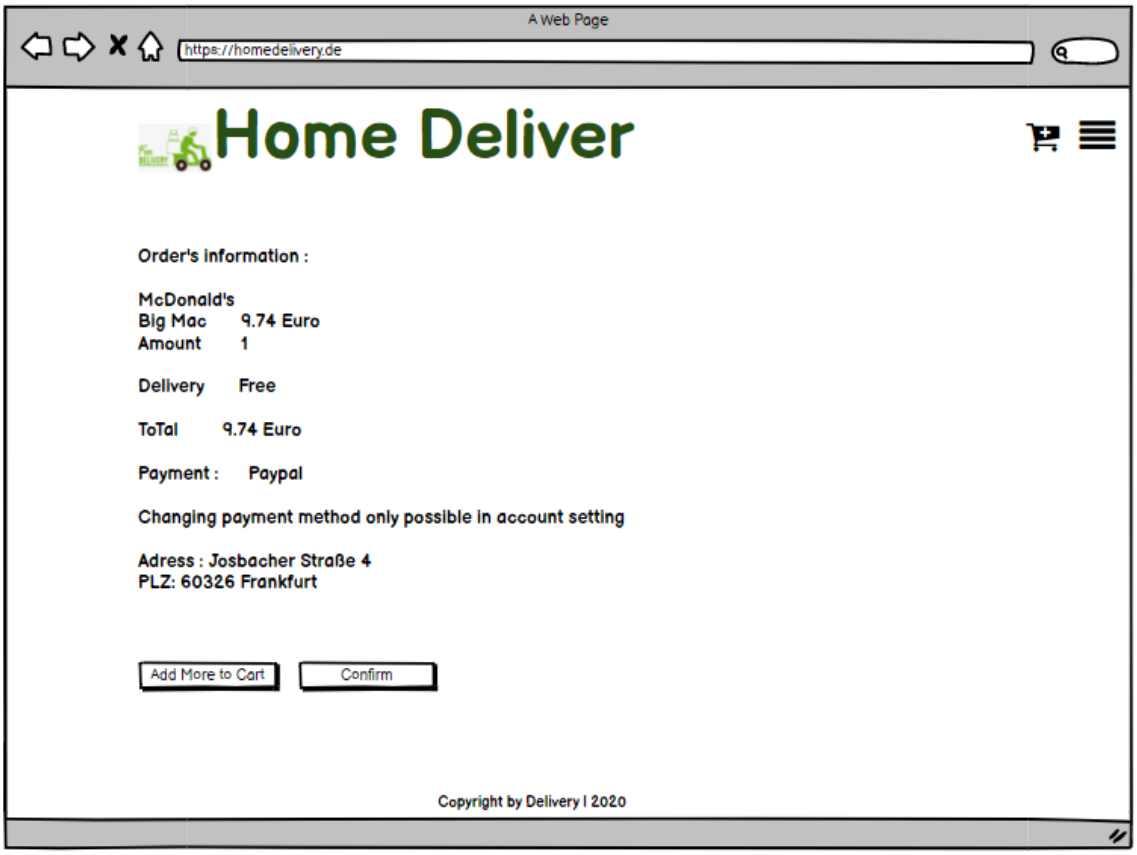


Figure 4.3: *Mobile GUI prototype oders information*

Figure 4.4: *Mobile GUI prototype conformation of order*

Categorization of requirement(functional/non-functional)

Functional requirement:

- Display all available products
- Saving selected products into shopping cart
- Change/remove items in shopping cart
- Data (time and costs of the trip) must be saved in our database

Non-functional requirement:

- User has to confirm his order

Business value of the corresponding functionality

We have to show customers our prices. We will provide the best options and a good transparency for them.
Therefore, in the user interface, the information of shipping costs will be shown. With the data we can know where the customers are, so that we can estimate the time correctly and send a shipper to the correct destination.

The whole list of available providers and products will give customer a very big and clear view of all items. They can freely have many choices. On the other hand, we can offer a very attractive marketing service.

Effort estimation for the implementation of the requirement

Our programmers will need to create a database for the shopping cart. Also, the remove/change in shopping cart has to interact in real-time to database.
Programmers also has to extract the data- order information of customers so that it will be sent to contractors.

UML Diagrams

Use case diagram:

Our main actor is user. An Order Confirmation is only in process if user is logged in and he has to confirm his Cart.

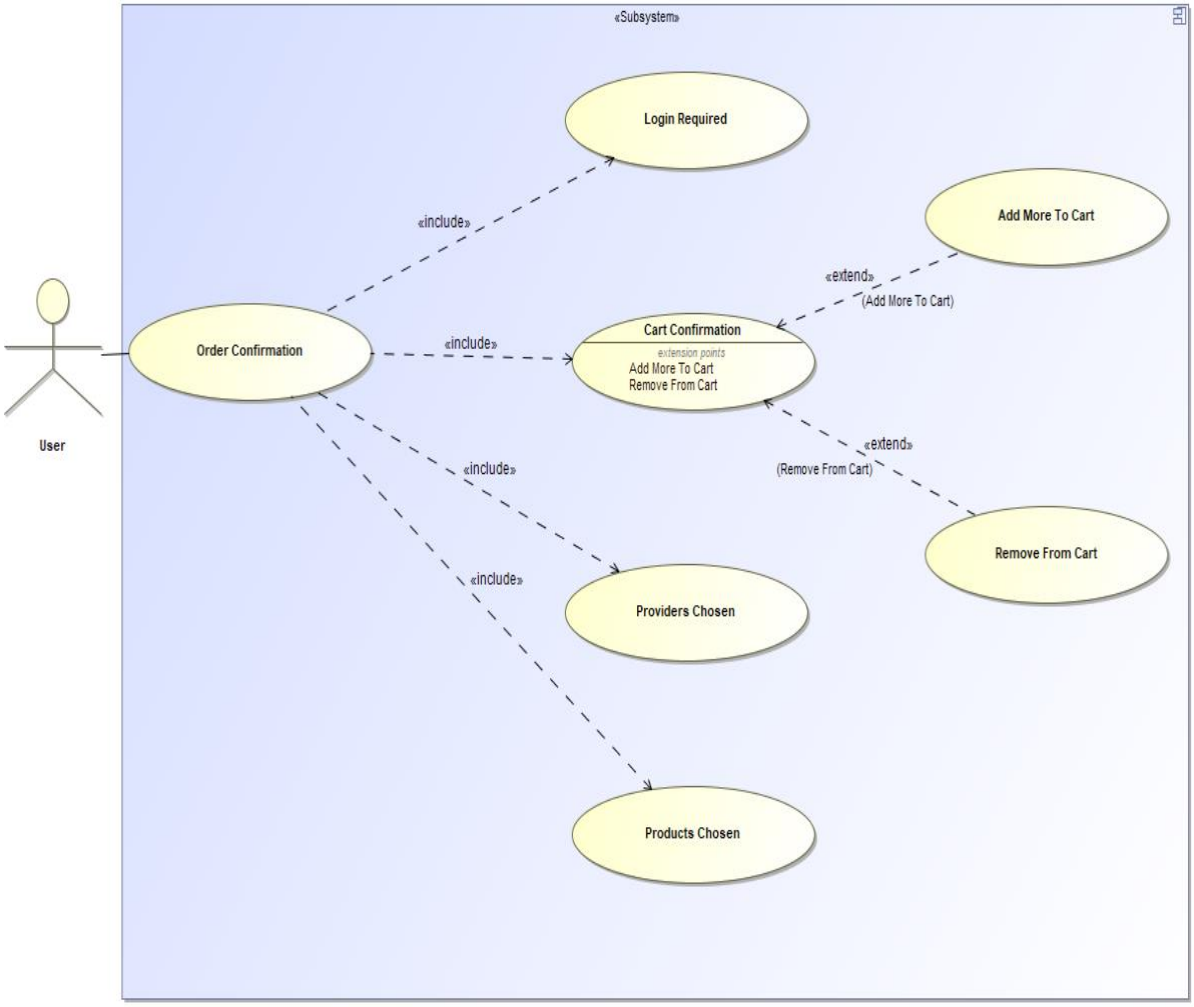


Figure 4.5: *Use Case Diagram*

Actor User

- User performs ConfirmOrder

UseCase OrderConfirmation

- OrderConfirmation is performed by User

- OrderConfirmation includes LogInRequired
- OrderConfirmation includes CartConfirmation
- OrderConfirmation includes ProvidersChosen
- OrderConfirmation includes ProductsChosen

UseCase ConfirmCart

- ConfirmCart extends AddMoreToCart
- ConfirmCart extends RemoveFromCart

Activity diagram:

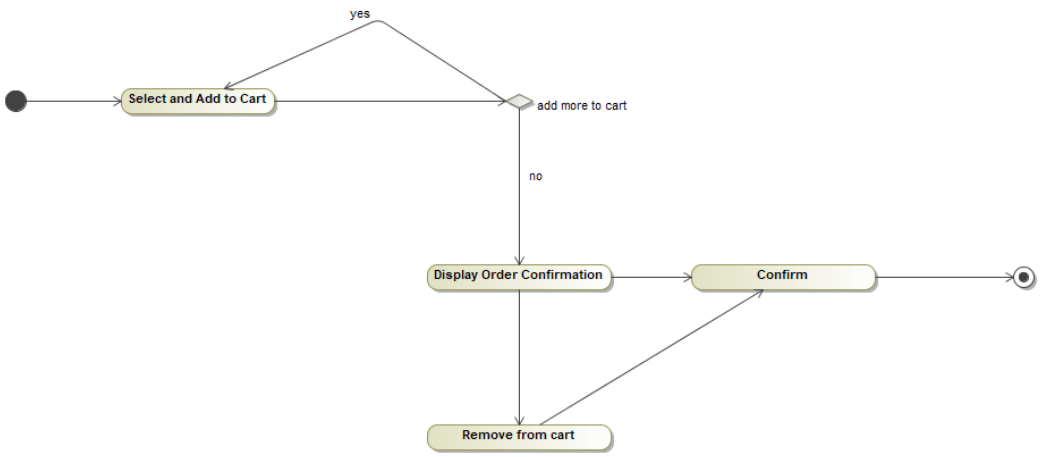


Figure 4.6: *Activity Diagram*

Sequence diagram:

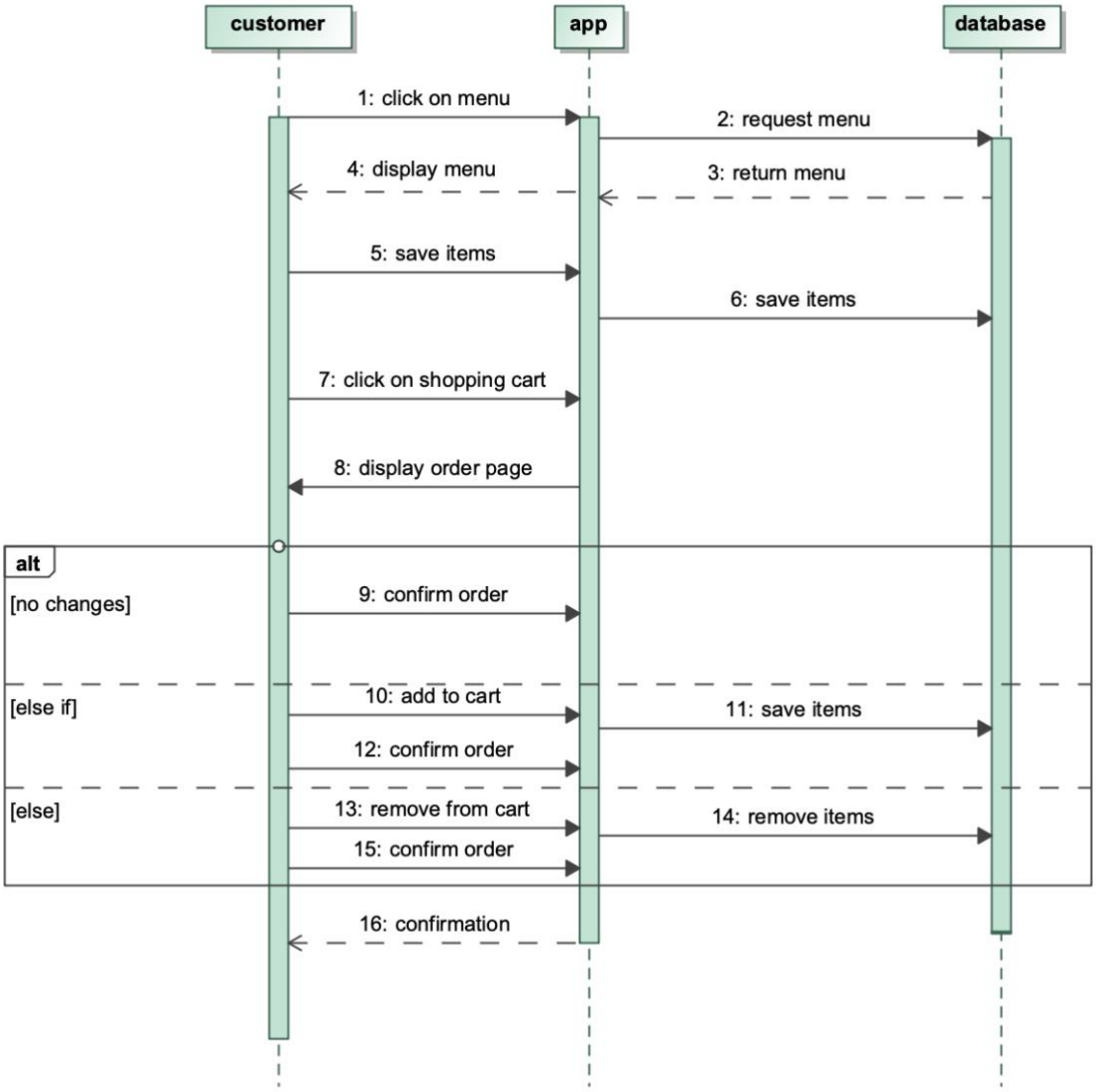


Figure 4.7: *Activity Diagram*

DA - Delete Account

As User: I want to be able to delete my account and be sure my data is deleted and won't be misused.	When the user deletes his account, it will be deactivated for 30 days if he logs in during this time his account will be reactivated if not all his data has to be deleted after 30 days. The data required for economic analytics has to be saved in a way that the user is untrack able.

Categorization of requirement (functional / non-functional):

- It is a functional requirement since the program allows deleting the data and user accounts.
- A non-functional requirement would be filtering the data, in which the program analyst selects what should be deleted and what not.

Business value of the corresponding functionality:

- It is very important to delete all of the user data if he deletes his account, because it's illegal to keep his non-needed data without his consent.

Effort estimation for the implementation of the requirement:

- We need a programmer that writes the program and analysts to distinguish the needed data from the non-needed one. It might take about three hours for the implantation.

For UI related functions: UI prototype

Figure 5.1: *Mobile GUI prototype delete account*

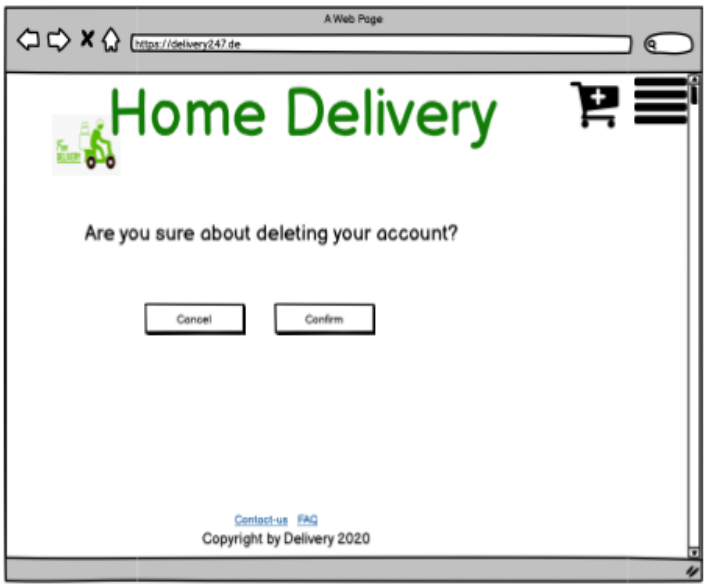


Figure 5.2: *WEB GUI prototype delete account*

UML Diagrams:

Use Case scope

The user wants to delete his account. He goes to setting and clicks on delete account. He then gets a pop-up notification, where the program asks him if he is sure about deleting his account, he can either cancel or confirm deleting his account. If he confirms, then his account will be deactivated for 30 days and the analyst will then analyze the data and delete it after the deadline. The data needed for business or economic purposes will be kept.

Use case diagram:

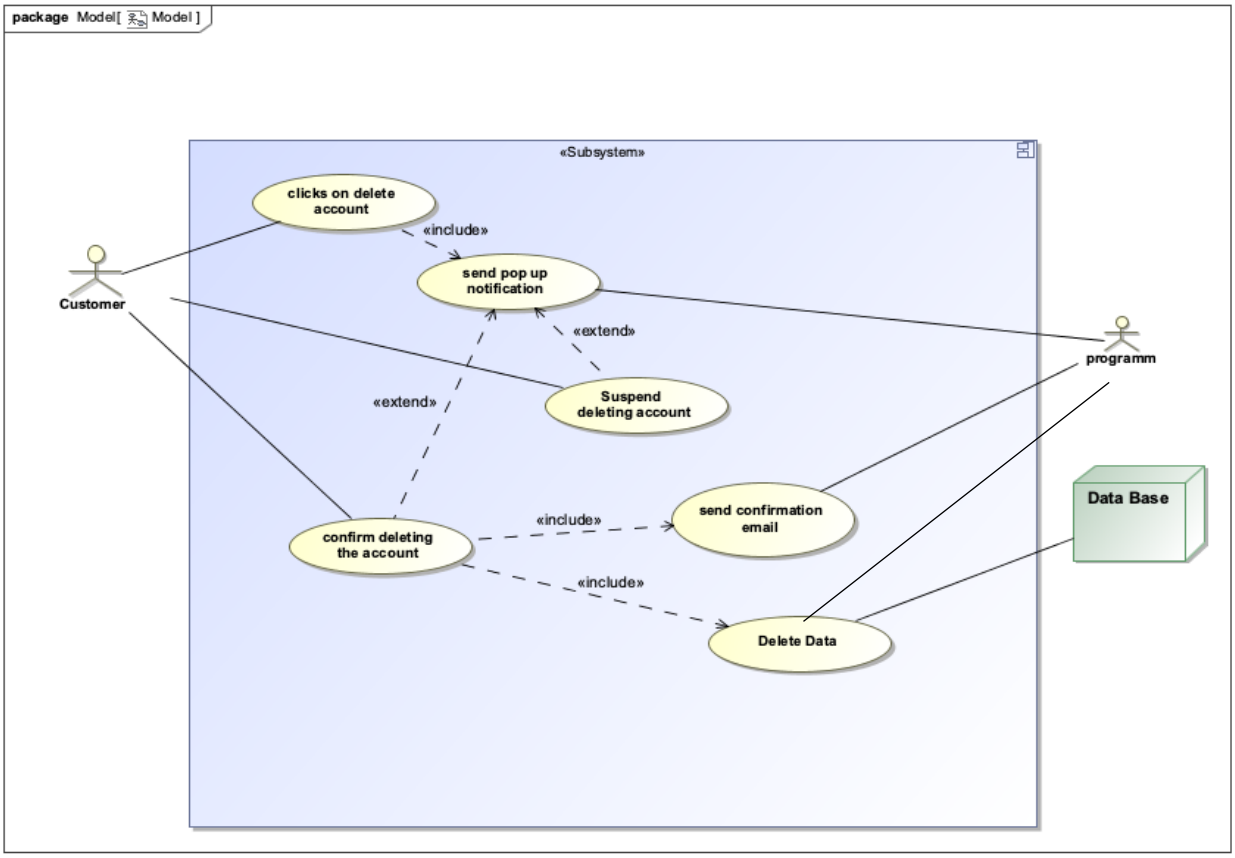


Figure 5.3: deleting account, use case diagram

Actor Customer

- Customer performs Clicks on Delete account
- Customer performs Suspend deleting account
- Customer performs Confirm deleting account

Actor Program

- Program performs Send pop up notification
- Program performs Send confirmation email

UseCase clicks on delete account

- clicks on delete account is Performed by Customer
- clicks on delete account includes Send pop-up notification

UseCase send pop up notification

- Send pop up notification is performed by Program
- Send pop up notification is included by Clicks on delete
- Send pop up notification is extended by Suspend deleting account
- Send pop up notification is extended by Confirm deleting account

UseCase Suspend deleting account

- Suspend deleting account extends Send pop up notification
- Suspend deleting account is performed by Customer

UseCase Confirm deleting account

- Confirm deleting account is performed by Customer
- Confirm deleting account extends Send pop up notification
- Confirm deleting account includes Send confirmation email
- Confirm deleting account includes Delete Data

UseCase Send confirmation email

- Send confirmation email is performed by Program
- Send confirmation email is included by Confirm deleting account

UseCase Delete data

- Delete Data is performed by Program
- Delete Data is included by Confirm deleting account

Class Diagram:
This requirement doesn't need a class diagram, because there are only actors that do actions and no object to put in a class diagram.

Activity Diagram:

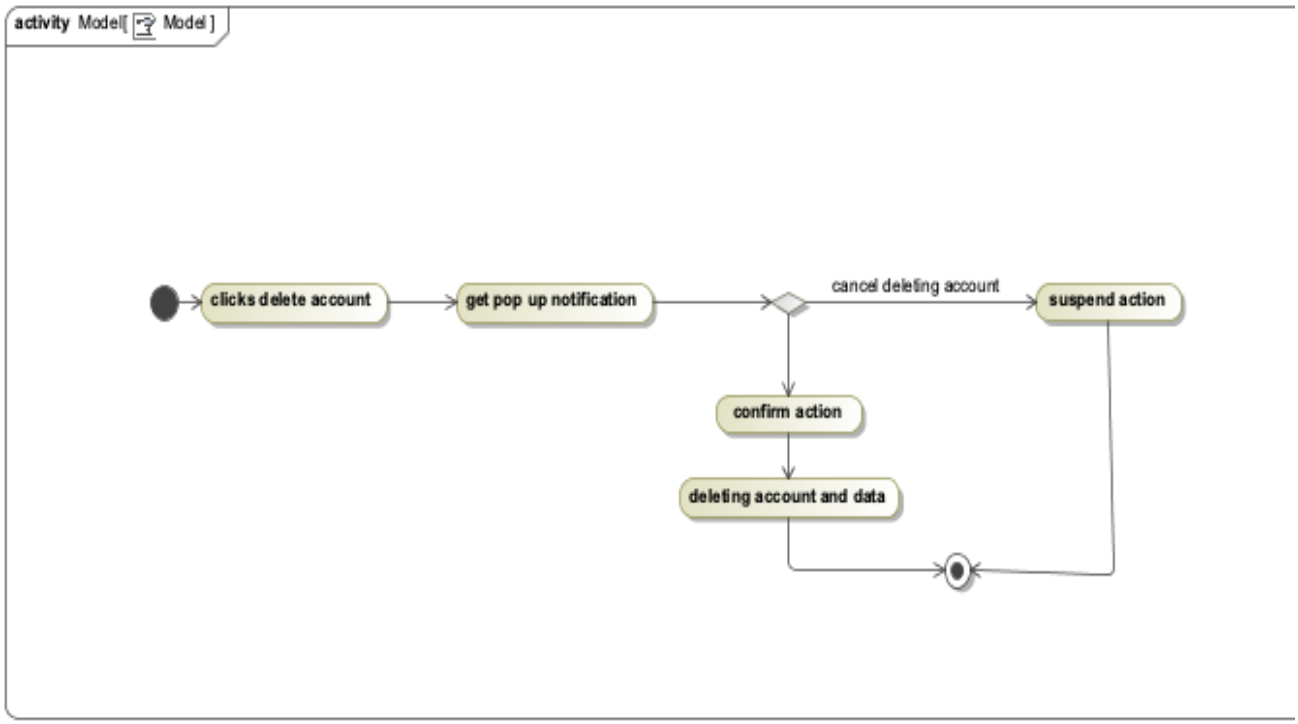


Figure 5.4: Deleting account, Activity Diagram

Sequence diagram:

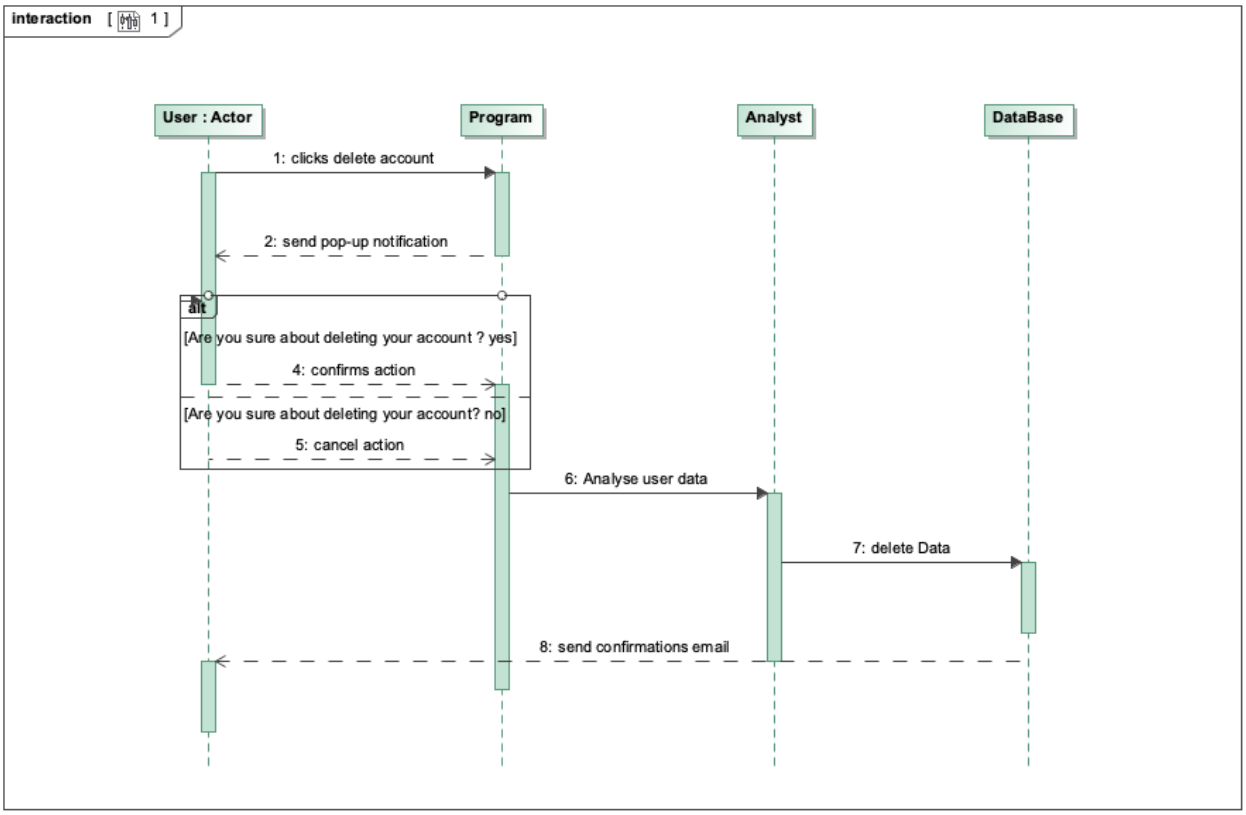


Figure 5.5: Deleting account, Sequence Diagram

RT – Rating System

Customer: After the contractor confirms the delivery the customer gets a notification to rate the service using stars' classification system, which he can either accept or skip. The customer is asked to rate the following aspects: -Friendliness -Time -State of the order	The customer is able to rate our App, or skip the rating.

Categorization of requirement (functional / non-functional):

Functional:

- the user can rate the App after getting his order

Non-functional:

- the server needs to collect the data entered by the customer
- Business value of the corresponding functionality:
- Being able to rate our App indicates that we care about our clients, and we do
- this in order to improve our services and also assure that we're doing our job well.

Effort estimation for the implementation of the requirement:
First of all, we need to program this function and an analyst to observe the rating activity of the customers, and delivering what should be improved (e.g. punctuality, friendliness or state of the order).

For UI related functions: UI prototype

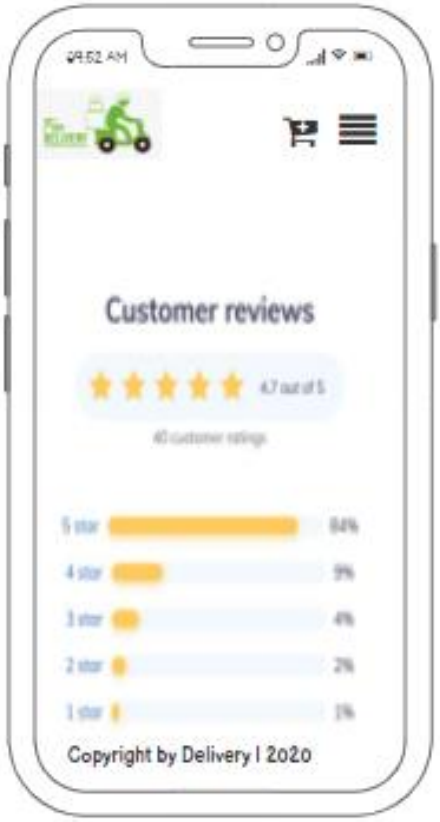


Figure 6.1: *mobile prototype RS platform*

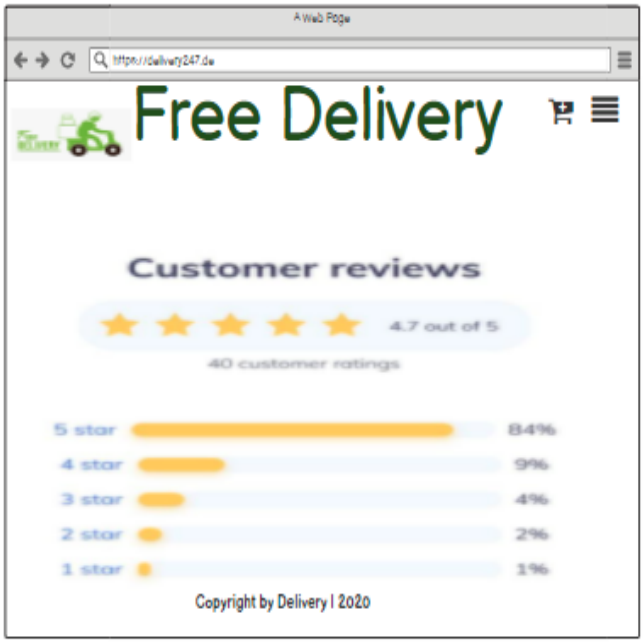


Figure 6.2: *Web prototype FAQ and Contact us platform*

UML Diagrams

Use case diagram:

Use case scope

If the customer receives his order, he gets a pop-up notification, where the program asks him if he wants to give a feedback about our application, he has the ability to skip the rate or to accept the rate, and rate our app from 1 to 5 stars for each of: friendliness of the contractor, the punctuality of the order and last about the state of the order.

Use case diagram:

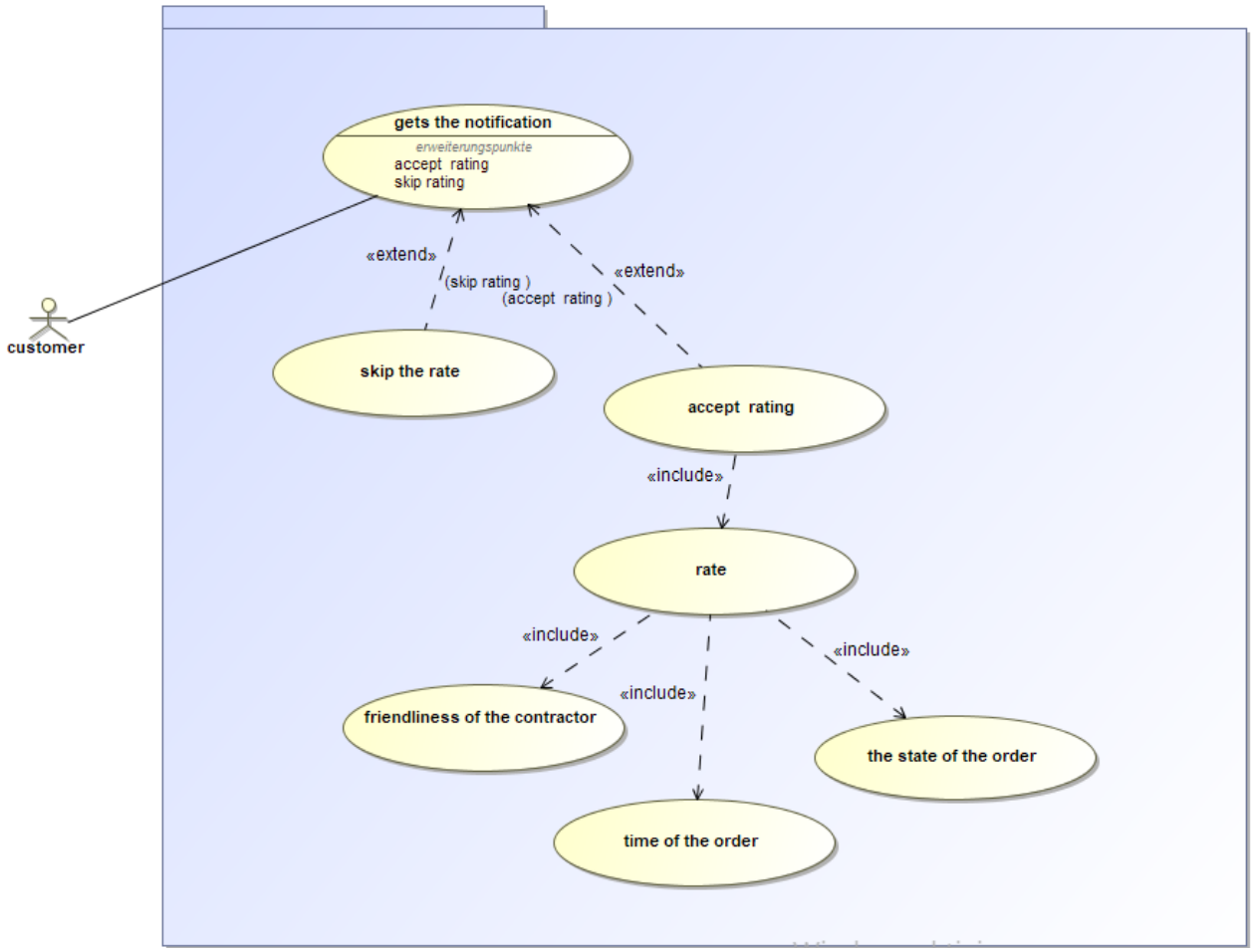


Figure 6.3: Rating system, use case diagram

Actor System:

- the system performs send notification

Actor Customer:

- Customer performs Skip Rating
- Customer performs Accept rating
- Customer performs Rating

useCase Send Notification:

- Send notification is performed by the system

useCase Skip rating:

- Skip rating is performed by customer
- Skip rating is included by Get notification

useCAse Accept rating:

- Accept rating is performed by customer
- Accept Rating is included by Get Notification

useCase rating:

- Rating is performed by customer
- Rating is included by accept rating

Class Diagram:

The class Diagram will not be needed in our case, and that's because we only have one actor.

Activity Diagram:

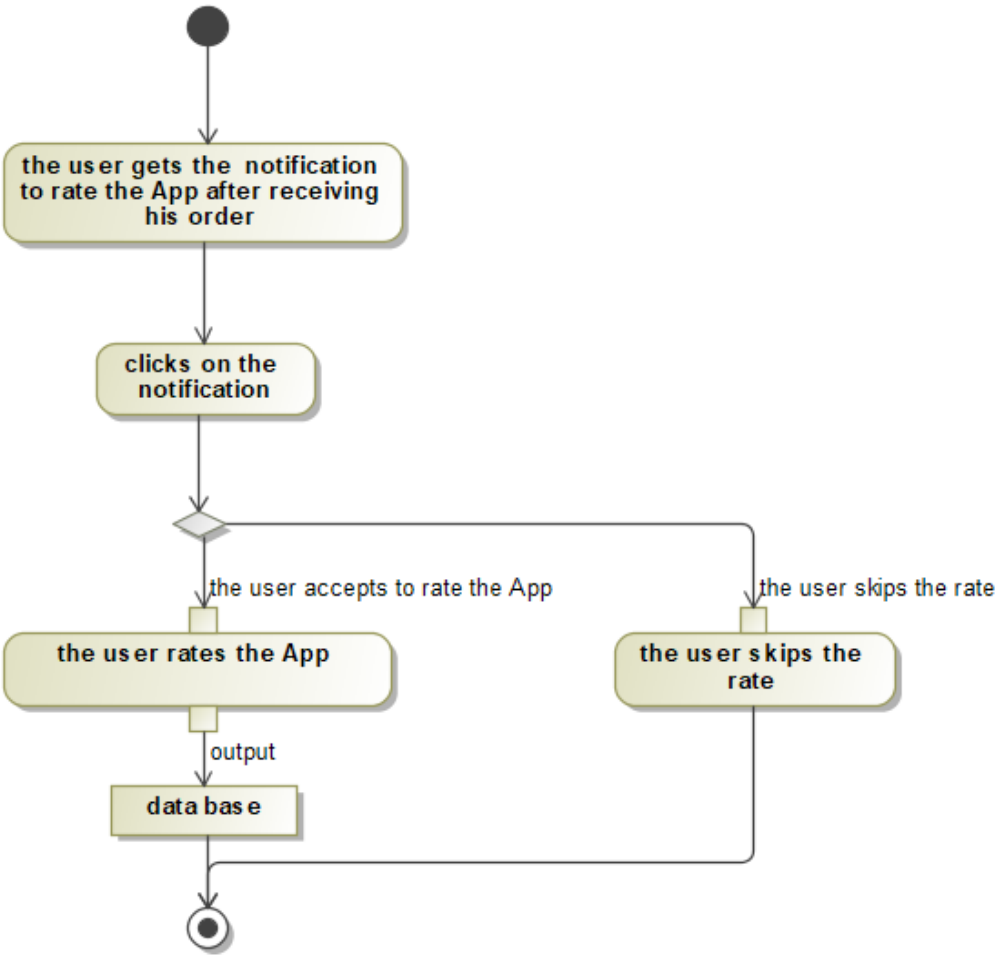


Figure 6.4: rating system, activity diagram

Sequence Diagram:

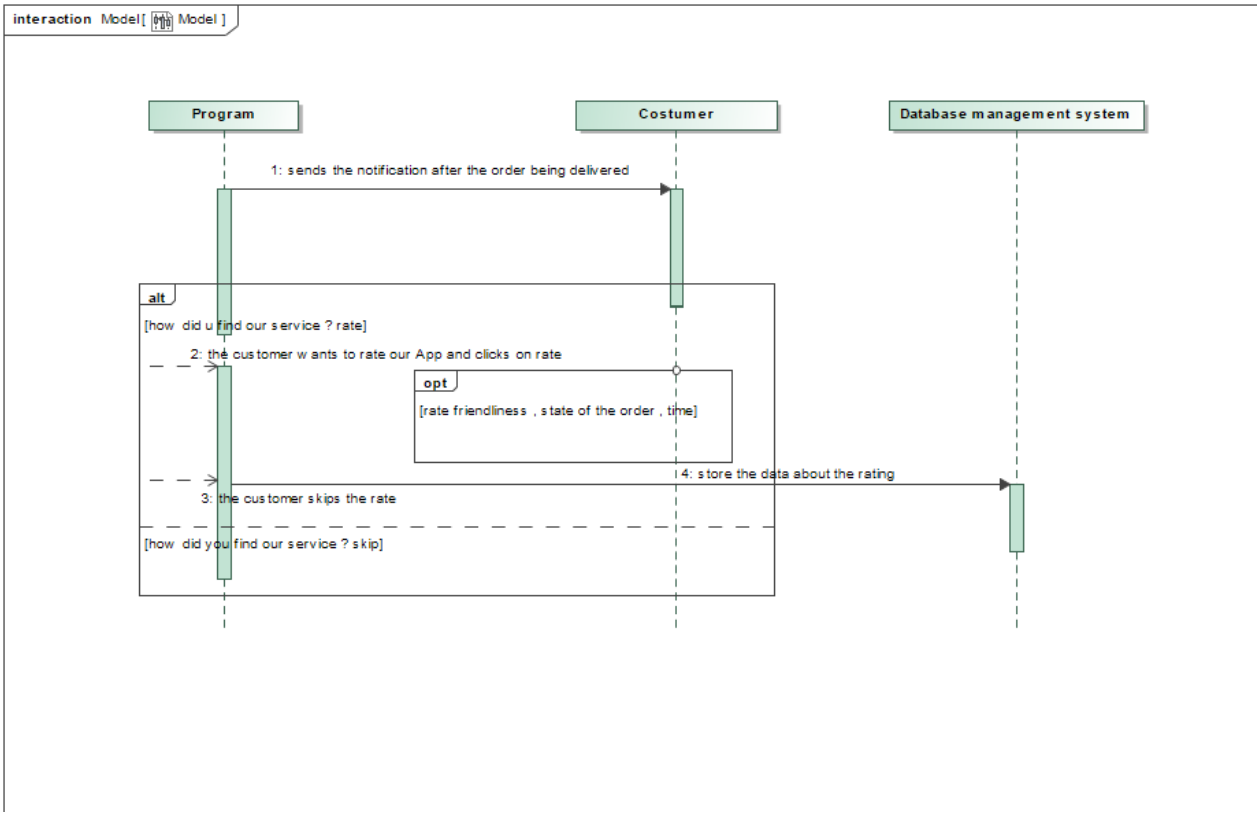


Figure 6.5: Rating System, sequence diagram

Retrospective #1

After our meeting we chose to split the retrospective into two parts : “What went well “ and “things that should be improved “ .

What went well?

- Everyone chose his requirement without any problem
- We helped each other with some suggestions

Things that should be improved?

- we should work on our communication skills
- understand our tasks because sometimes we did the job twice because of misunderstanding problems

Sprint planning #2

The role of the product owner will be taken by Tung Trinh in the second sprint. We re-estimated the backlog items together and added some requirement we found necessary.

The following table shows tasks distribution of the second sprint which begins on the

ID	Assigned to	Begin	Finished	Workload in each item in percent
RAM4	Ceylan Yildirim	16.01.2021	23.01.2021	100%
OPC1	Tung Trinh	16.01.2021	23.01.2021	100%
OPC2	Tung Trinh	16.01.2021	23.01.2021	100%
FACP	Lamiaa Kamar	16.01.2021	Removed to sprint #3	100%
GRS	Nouhaila ben Messaou	16.01.2021	Removed to sprint #3	100%
SAA	Nouhaila ben Messaou	16.01.2021	Removed to sprint #3	100%

Table 2: Task distribution of the second sprint, sprint backlog #2

Use Case Description

Use Case field	**The customer can change his password.**
Use Case name	Customer / Contractor registration and management
Subject area	Customer / Contractor registration and management
Trigger	User: • Click on change password
Actors	User
Use case overview	The Customer have to enter his password and the new password.
Preconditions	Enter current password.
Termination outcome	A positive ending: User successful changes his password.

	A negative ending: User enters false password and can't change password.
Use Case description	User can change his password.
Use Case association	User
Input summary	Current password, new password
Output summary	Password changed
Use case notes	None

Use case description : *Use case description for RAM4*

Use Case Field	**A list of deliveries is shown to contractors.**
Use Case Name	Order PickUp by Contractor
Subject Area	Order PickUp by Contractor
Trigger	Contractor: • Click on list of works
Actors	Contractor
Use case overview	An overview of all deliveries
Preconditions	Contractor is logged in
Termination outcome	A positive ending: Always in time delivery. A negative ending: Late delivery.
Use Case description	He logs in to see the list of deliveries to be done. Order will be delivered base on the displayed time.
Use Case association	Contractor,Database
Input summary	The contractor has to do all the works given
Output summary	Different deliveries
Use case notes	Order information , arrived time has to be displayed in the end

Use case description: for OPC2

Sprint review #2

Item reviews performed by Ceylan Yildirim:

RAM4:

- Changing password via email changed
- Changing password only possible in the setting if the customer is logged in
- Current password is required

New passwords have to be entered with min. 8 characters (letter, special character and numbers)

Item reviews performed by Tung Trinh:

OPC1 and OPC2:

- estimate the effort required to implement the requirement
- sequence diagram is hard to implement

Results of sprint #2

RAM4: Changing password

As User / Contractor Forget password	New password request per e-mail. You can change your password by clicking on the link in the email.

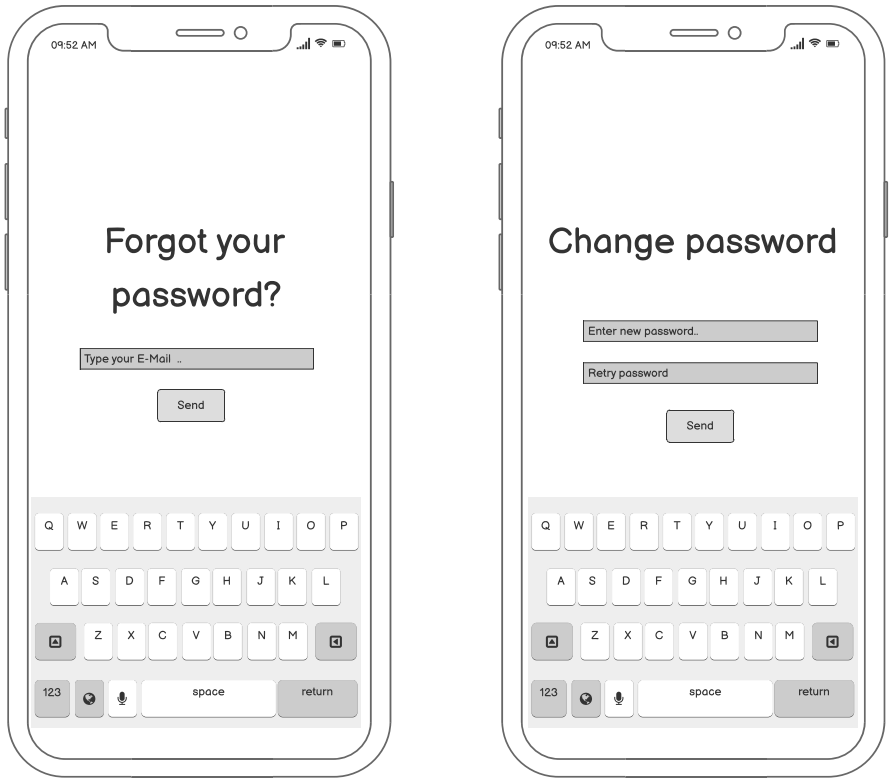


Figure 7.1: *Mobile GUI prototype password changing*

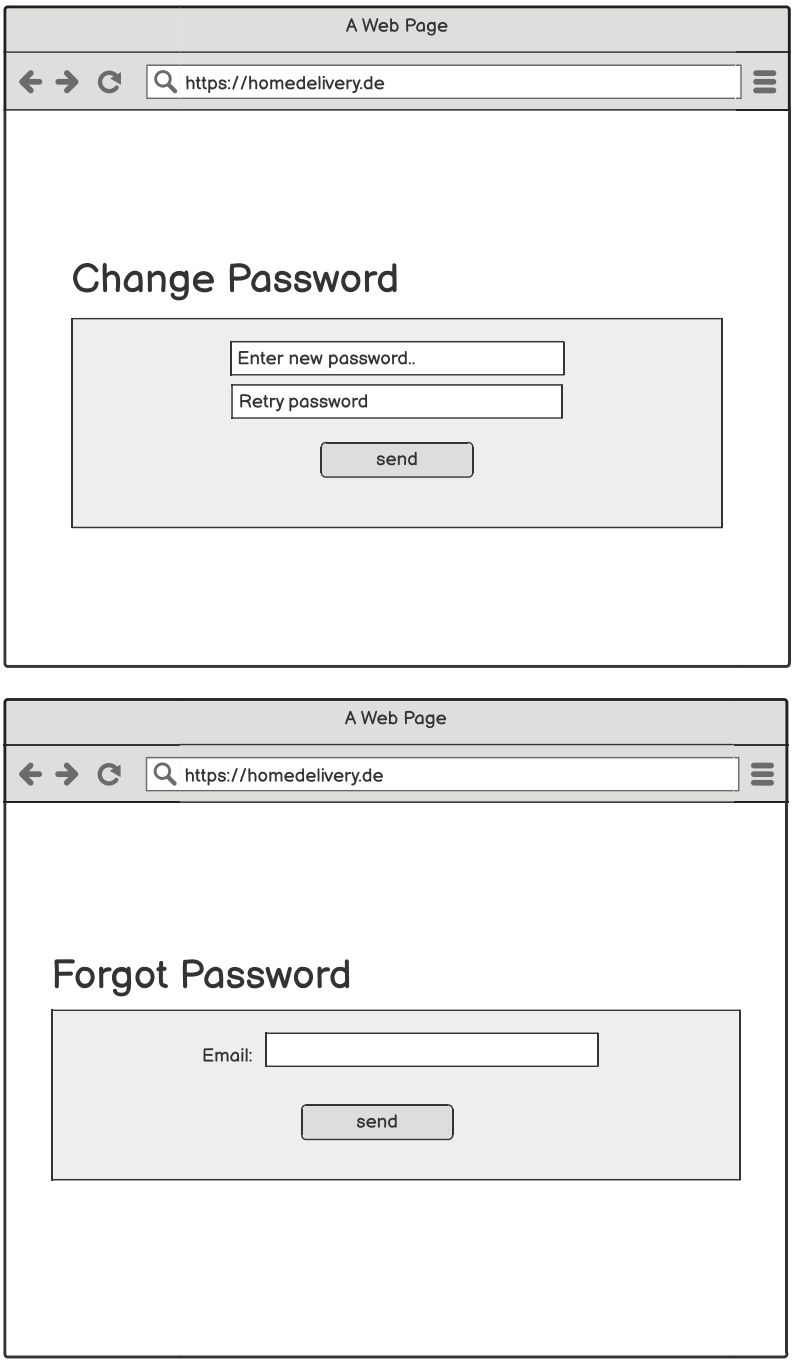


Figure 7.2: *WEB GUI prototype changing password*

UML Diagrams:

Use case Diagram:

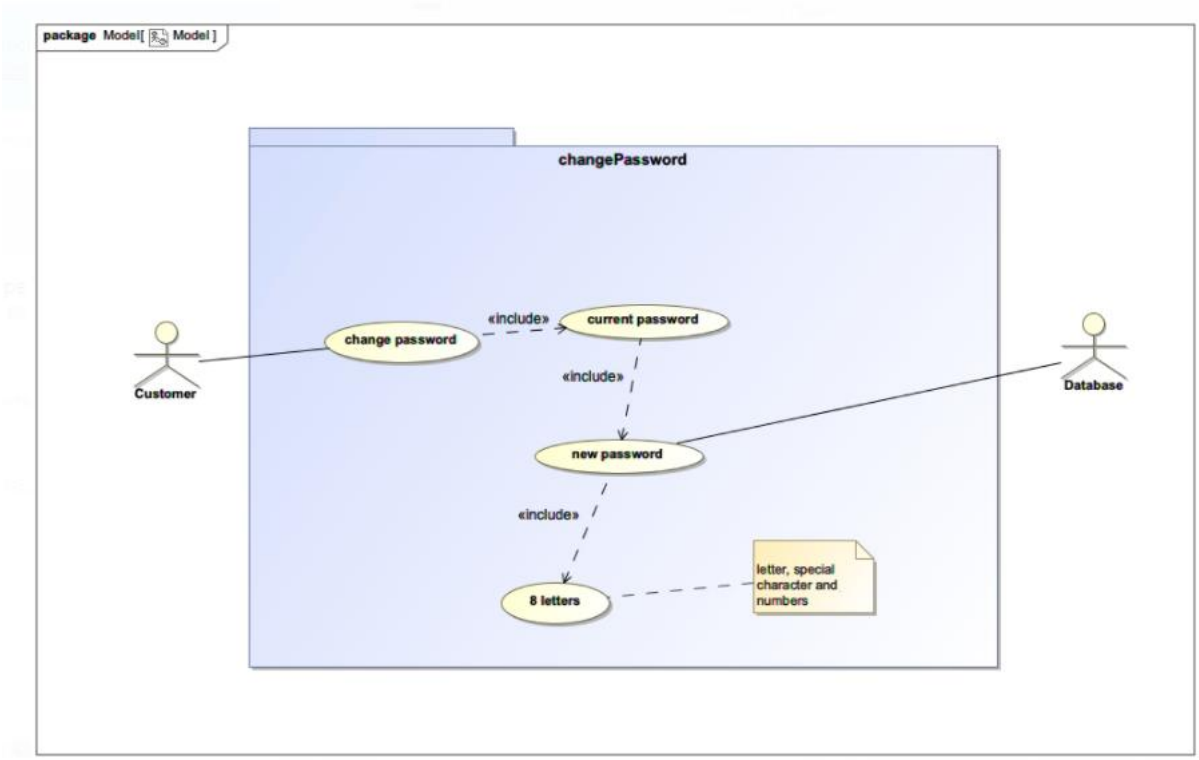


Figure 7.3: *Use Case Diagram*

Class Diagram:

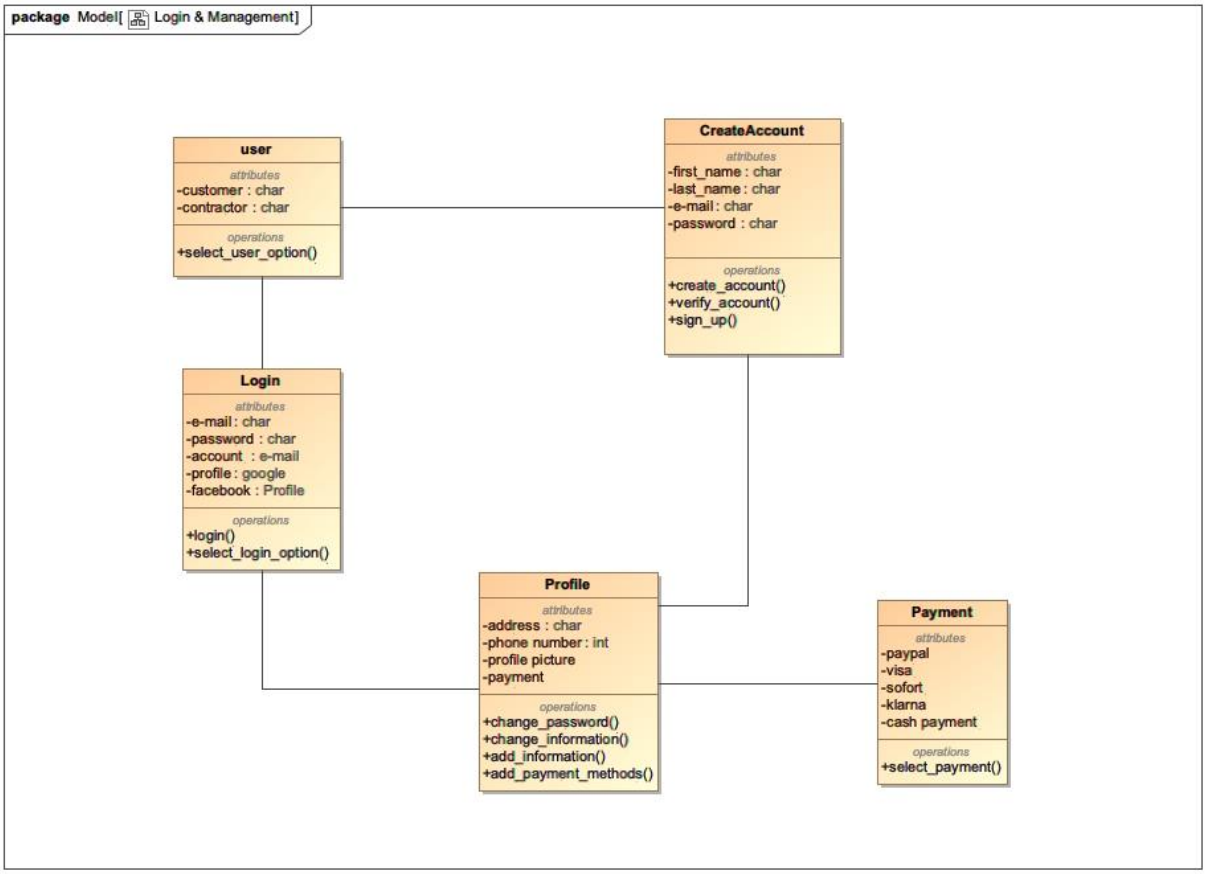


Figure 7.4: *Class Diagram*

Activity Diagram:

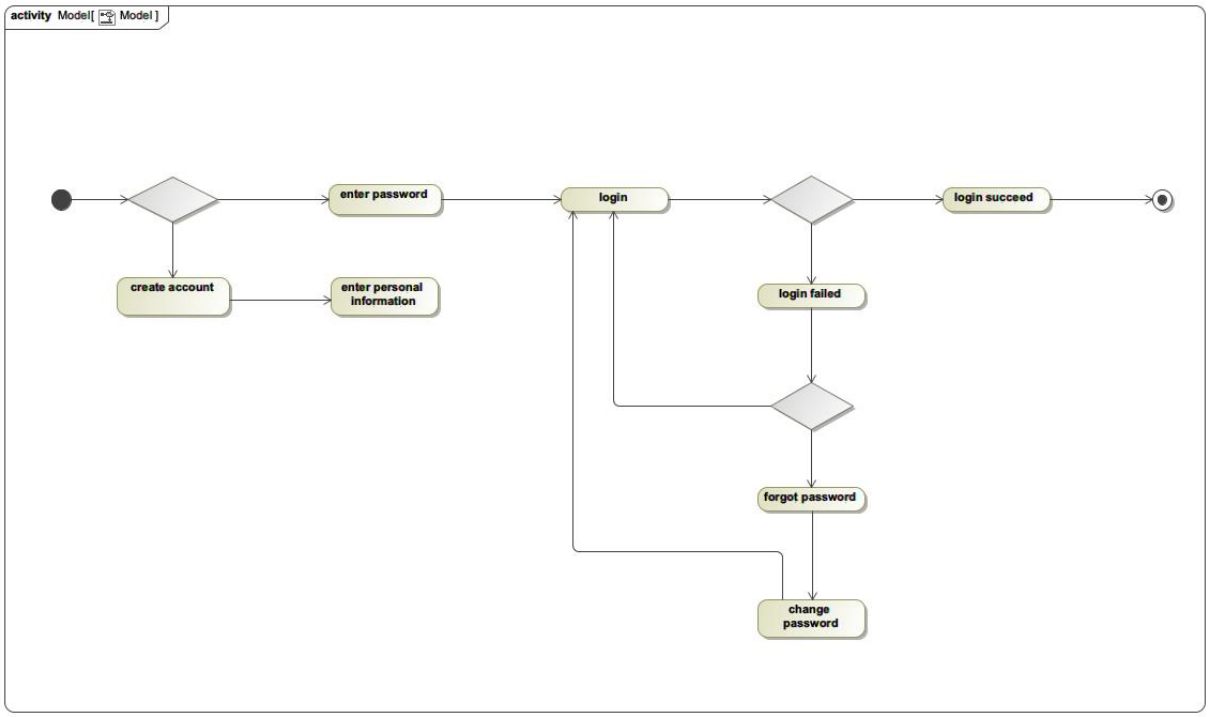


Figure 7.5: *Activity Diagram*

OPC1 – Order Pickup by Contractor

As Admin I will decide who receives which Order.	An Algorithm will be used to choose who will take which order. Contractors will be select after these priorities: distance, activity status and rating reviews.

Categorization of requirement(functional/non-functional)

Functional requirement:
Algorithm to sort and select contractors

Non-functional requirement:
Order has been confirmed.

Business value of the corresponding functionality

Many companies are working on this field. In order to compete with them, we need an innovative idea. We will only hire permanent employee.

With these ideas of business, we will be able to bring the best service to customer. Their orders will always be accepted and delivered.

On the other hand, contractors will have the almost same number of orders as the others. This would bring a balance to our business.

Effort estimation for the implementation of the requirement

We need our programmers to implement an algorithm to sort and select contractors.

Priorities will be distance and activity status. The algorithm will pick up the nearest contractor, only if he is free at that moment. If not, the algorithm will jump and look up for the next nearest person.

In the end, the algorithm should give each contractor an equal (not exact) amount of orders.

In order to implement this algorithm, we need experienced programmers. The algorithm should give a correct result within seconds so that the contractors are informed in real time.

UML diagrams:

Use Case Diagram:

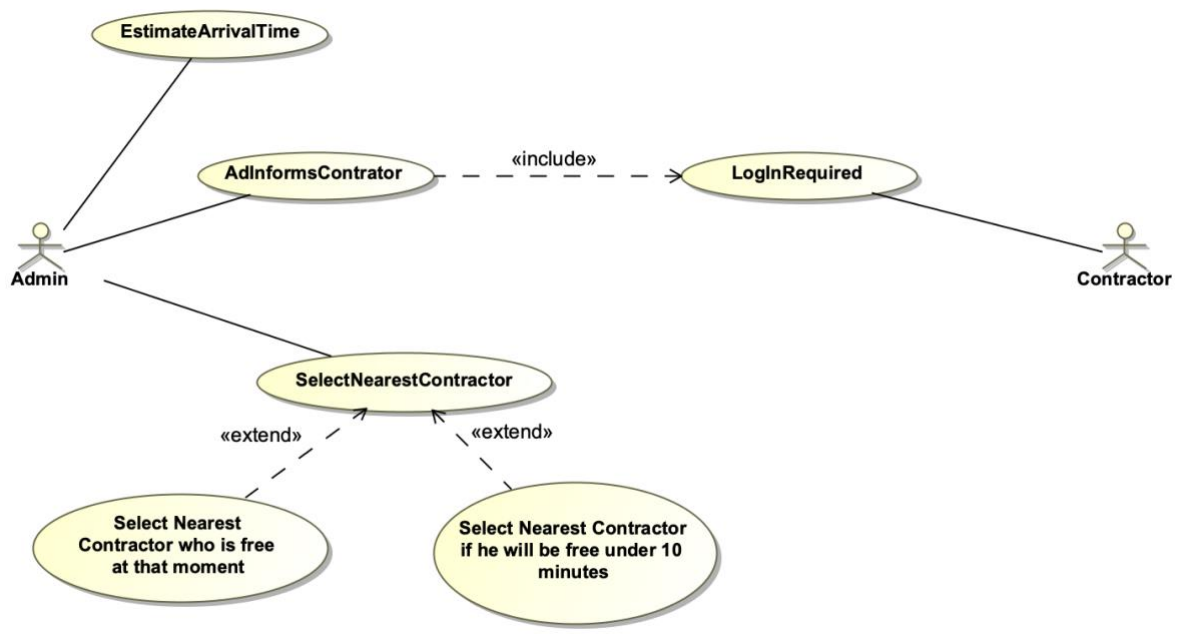


Figure 8.1: *Use Case Diagram*

Actor (Admin)

- Admin performs EstimateArrivalTime
- Admin performs AdInformsContractor
- Admin performs SelectNearestContractor

Actor (Contractor)

- Contractor performs LoginRequired

UseCase AdInformsContractor

- AdInformsContractor includes LogInRequired

UseCase SelectNearestContractor

- SelectNearestContractor extends SelectNearestContractorWhoIsFreeATM
- SelectNearestCOntractor extends
- SelectNearestContractorIfHeWillBeFreeUnder10Minutes

Activity Diagram:

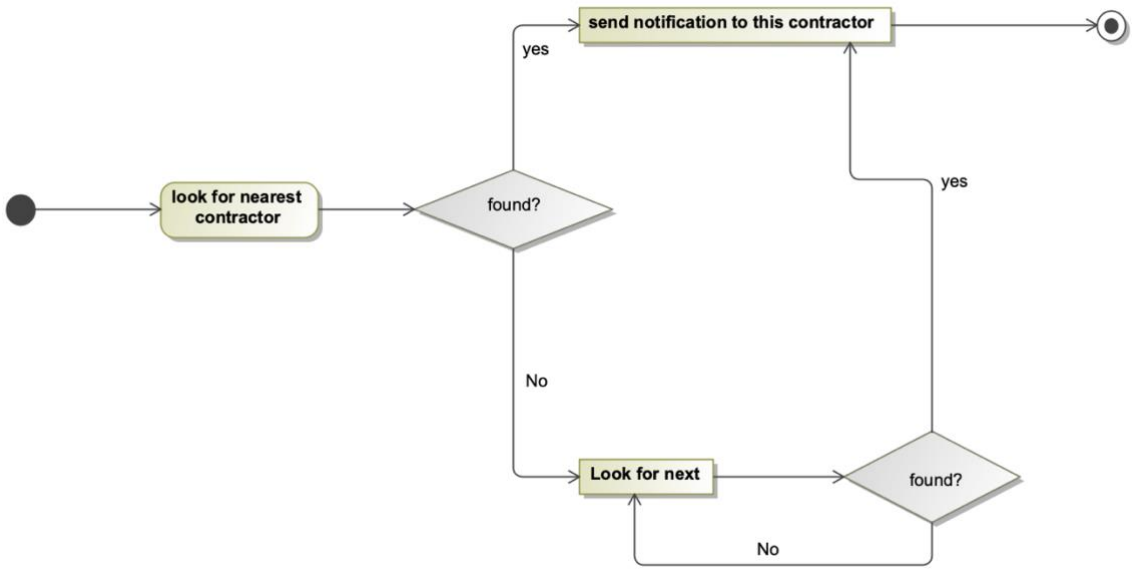


Figure 8.2: *Use Case Diagram*

Sequence diagram:

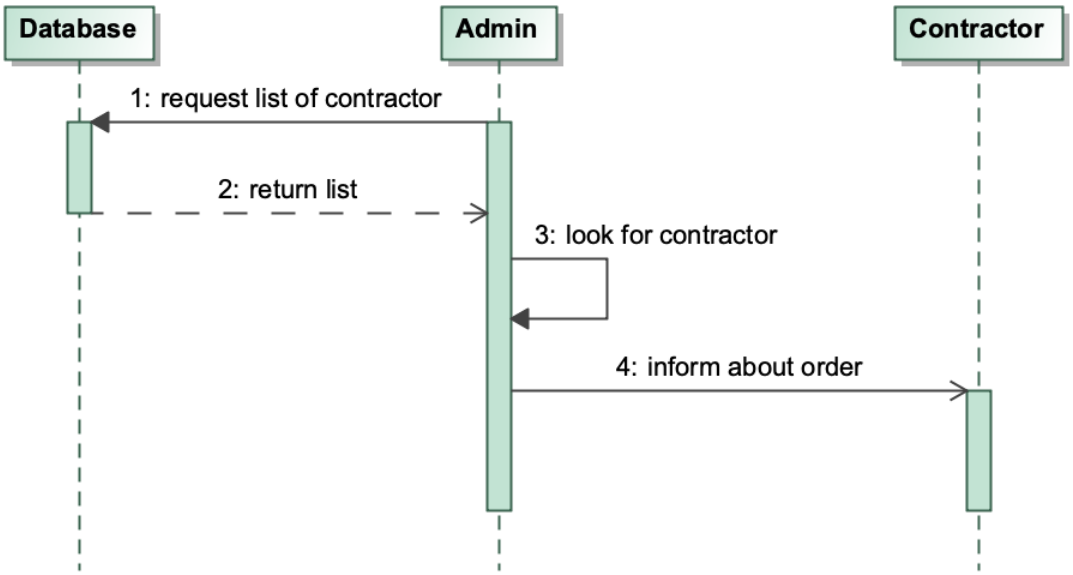


Figure 8.3: *Sequence Diagram*

OPC2 : Contractor's perspective

As contractor, I will be informed by the app. My job is only to go to the provider, pick up the order and deliver it the customer in time.	The app gives me notifications about the order information. Very important, I have to deliver orders in time. The time will be calculated from the app and it will be displayed for both contractor and customer.

For UI related functions: UI prototype

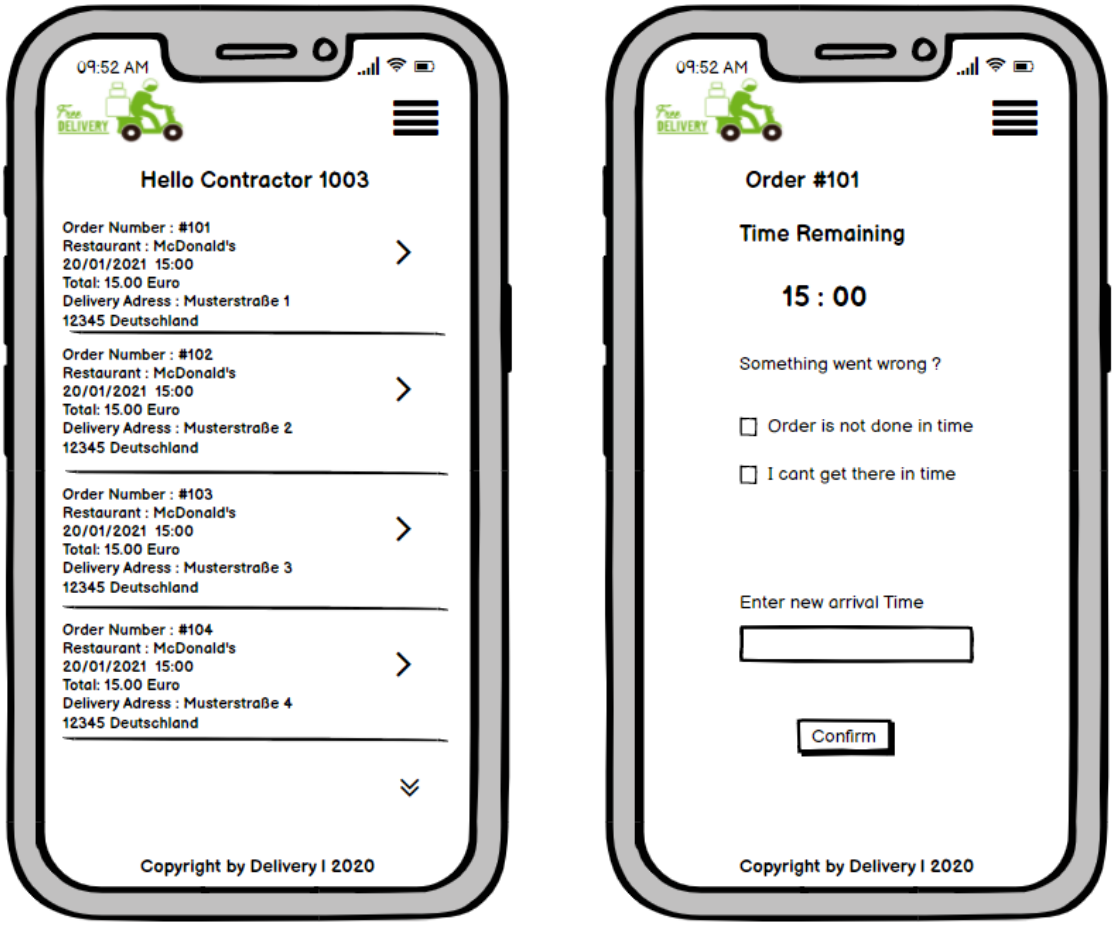


Figure 9.1: *Mobile GUI prototype time remaining*

Categorization of requirement(functional/non-functional)

Functional requirement:

- Display Order Information
- Display Provider Location
- Display Customer Location
- Display Delivery Time

Non-functional requirement:

- Order has been confirmed.

Business value of the corresponding functionality

As I already mentioned above, contractors will get the almost same numbers of orders at the end of the day. They don't need to fight or compete for getting orders. Their jobs are to delivery in time and be friendly to customer.

UML Diagrams:

Use Case Diagram:

Our Actor is the delivery boy.

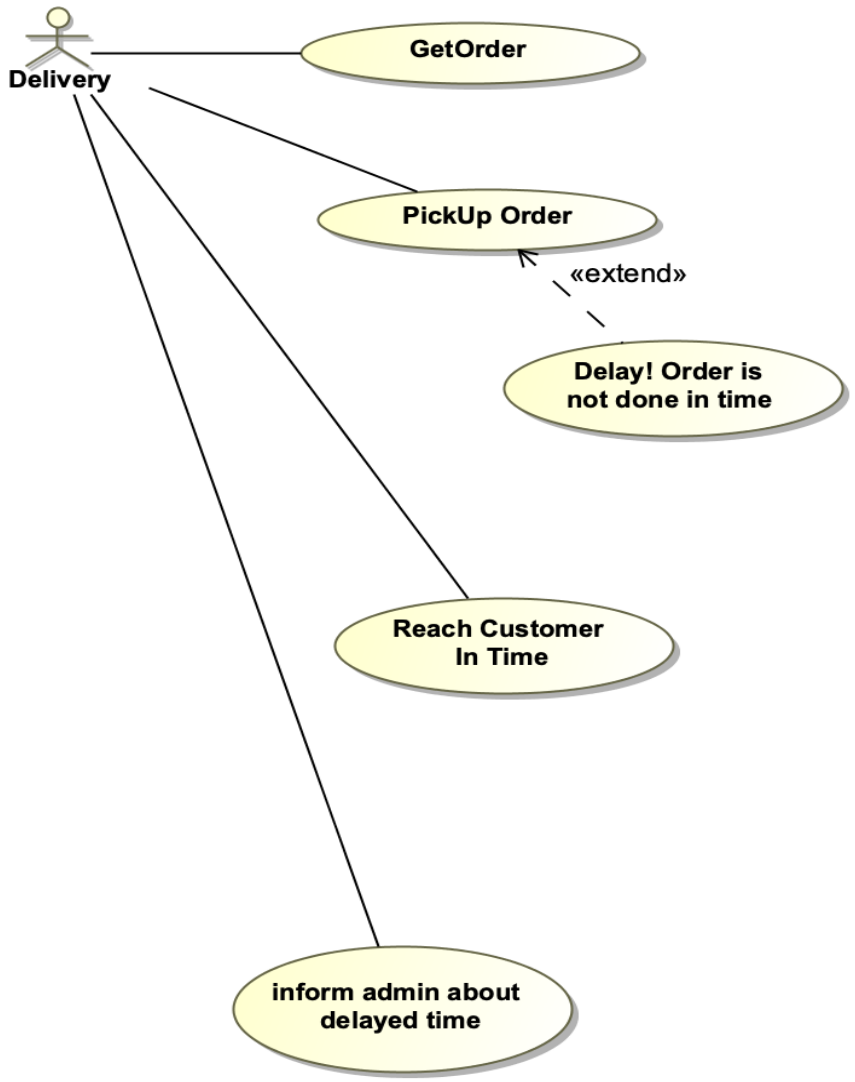


Figure 9.2: *Use Case Diagram*

Actor (Delivery)

- Delivery performs GetOrder
- Delivery performs PickUpOrder
- Delivery ReachCustomerInTime
- Delivery InformAdminAboutDelayedTime

UseCase PickUpOrder

- PickUpOrder extends Delay

Activity Diagram:

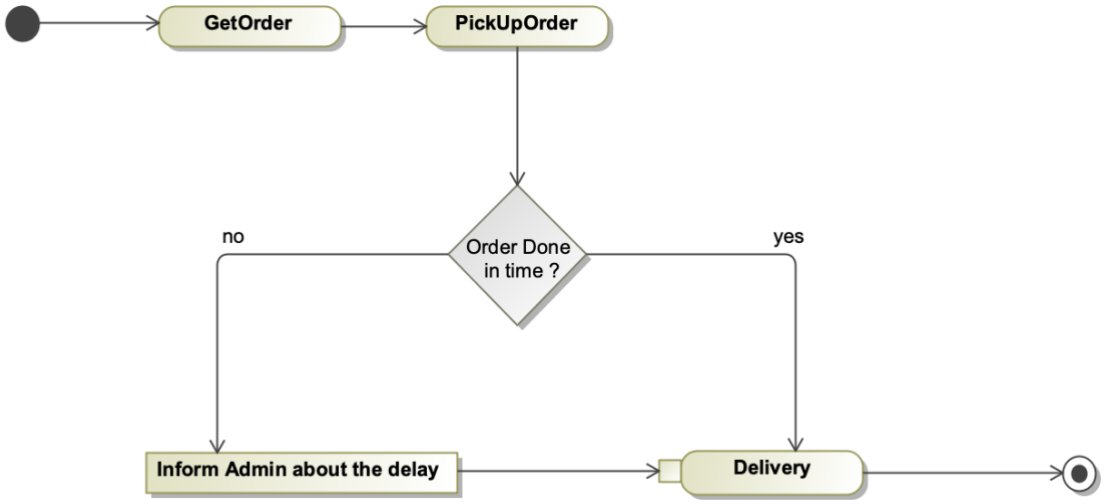


Figure 9.3: *Activity Diagram*

Sequence Diagram:

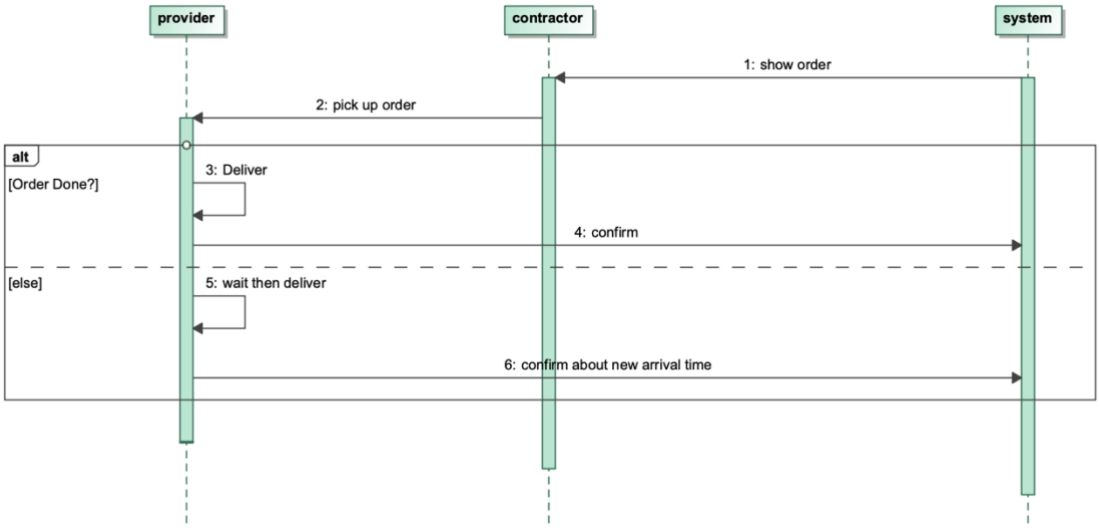


Figure 9.4: *Sequence Diagram*

Retrospective #2

After our meeting we chose to split the retrospective into two parts : “What went well “ and “things that should be improved “ .

What went well:

- We hold every week our meetings.
- Our tasks are distributed through Scrums.
- Everyone got the idea of the order confirmation.
- Communication takes place most of the time. No one is left alone.
- We helped each other.
- The tasks were always being done in time.
- We show our algorithms implemented for the tasks.
- Too many ideas were provided. We chose the best way for our project.

Things that should be improved:

- We could have done it better.
- It took us a while to figure out how to draw the diagrams correctly.
- We messed up the structure of our project.
- Too many ideas were provided.

Sprint planning #3

The role of the product owner will be taken by Nouhaila Ben Messaou in the third sprint. We re-estimated the backlog items together and added some requirement we found necessary.

ID	Assigned to	Begin	Finished	Workload in each item in percent
FACP	Lamiaa Kamar	16.01.2021	31.01.2021	100%
GRS	Nouhaila ben Messaou	16.01.2021	31.01.2021	100%
SAA	Nouhaila ben Messaou	16.01.2021	31.01.2021	100%

Table 3: Task distribution of the third sprint, sprint backlog #3

Use Case Description

The following table shows the task distribution for the third sprint:

Use case field	**Our Users need a platform where they can ask their questions and be in touch with the customer service and get adequate help.**
Use case name	FAQ and Contact-Us platform
Subject area	Customer service
Trigger	The user has a question→goes to settings→click on FAQ then on Contact us and he will be directed to it
Actors	User , customer service

Use case overview	if still no match in the FAQ the user can also get in touch with the customer service using the new implemented Contact-Us form.
Preconditions	The question has to be related to our delivery services.
Termination outcome	Positive ending: The User gets a useful answer and is satisfied with our support and continues using our App and recommends it to others. Negative ending: The User's question isn't answered or the platform doesn't function properly because of bugs which leads to user's dissatisfaction that way we lose the customer to other competitive company and get bad reputation
Condition affecting termination outcome	The stability of the internet connection and the staff members, who are responsible for answering the questions.
Use case description	The customer has a question concerning our services, if it's not already answer he can reach out to the customer service.
Use case associations	Settings menu
Input summary	The user's question
Output summary	Answers
Usability index	4
Use case notes	Most asked questions are listed on the platform to avoid repetition and time for both the user and the customer service employee and spare him answering high demanded questions several times.

Use case field	Our costumer gets rewards after ordering in our App, and being able to get them after a successful order.
Use case name	Reward System
Subject area	Customer satisfaction
Trigger	Order
Actors	The customer
Use case overview	The customer gets rewards after every order
Preconditions	Ordering from the App
Termination outcome	Positive ending: the customer gets his suitable reward

	Negative ending: the customer gets the false reward because of a problem in our algorithm or not getting the reward due to a bug in our system
Condition affecting termination outcome	A nonworking reward algorithm
Use case description	After each successful order, our customer gets reward points and keeps collecting them till he reaches the first discount which Is -5%, also if our customer orders 7 times , gets the 8th one for free.
Use case associations	
Input summary	Enter the gained points
Output summary	Suitable discount
Usability index	4

Use case field	**the company uses publicity for the marketing under one condition which is sufficient budget**
Use case name	Advertising
Subject area	External requirement
Trigger	None
Actors	The company
Use case overview	As a company we chose to use advertising which includes modern advertising for the people who social media , and classic advertising such as advertising panels , tv and radio, to increase our costumer circle depends on our budget to attract more people
Preconditions	Have enough budget to invest it
Termination outcome	Positive ending: people discover our App through our advertising methods and our costumer members increase with time.

	Negative ending: stability of the orders which means our advertising methods are not good enough.
Condition affecting termination outcome	Insufficient budget
Use case description	In case the revenues are higher then cost of advertising then we can invest them in Ads
Use case associations	The bank
Input summary	Money
Output summary	Advertising
Usability index	4

Sprint Review #3

Item reviews performed by Lamia Kamar:

FACP:

- The Contact-us form can also be used to leave comments, complaints or improvement suggestions.
- In the FAQ platform GUI add copyright
- Explain the missing class diagram

Item reviews performed by Nouhaila Ben Messaou:

RT:

- 5 points after each order
- The 8th order would be free
- 100 points: -5%, 200 points: -10%, 300 points: -15%, 500 points: -20%

SAA:

- Calculating the budget
- The revenue must be higher than the advertising budget
- Investing in both modern and classic advertising

Results of sprint #3

FACP - FAQ and Contact us platform

As a user I want to get competent assistance if faced by problems and be able to reach customer service for questions I can't find answer to or leave comments.	The customer service will be organized. The FAQ webpage will be organized and there shall be employees assisting and answering customers questions asked via contact us form.

Categorization of the requirement (functional / non-functional)

This requirement is functional because the program provides the functionality of helping users

Still non-functional requirements are included such as the quality of the system and its stability. Also usability should be very easy so that all users can use the program

A working webserver and a database are two non-functional preconditions

Business value of the corresponding functionality

The business value of this functionality isn't that high as other functionalities, yet by implementing a FAQ platform we can keep up with other companies' competition; since by affording enough support to user he might feel welcomed and may continue using our services or even recommend it to others.

Effort Estimation for the implementation of the requirement

The complete implementation of this requirement may take up to two days and needs to be updated throughout the business process.

For UI related functions: UI prototype

In the following graphical user-interface pictures our FAQ platform and contact-us form, which we include in our app and website.

Figure 10.1: *mobile prototype FAQ and Contact us platform*

Figure 10.1: *WEB prototype FAQ and contact us form*

UML Diagrams

Use Case Scope

The FAQ and Contact-us platform is implemented in order to help users get answers to their questions concerning our services and give them a mean to reach us.

The objective is to enhance the user's experience and offer him a good customer service that serves his needs, so they feel welcomed and close to the stuff and that way they would recommend the app to other potential future users, and that would help us with the competition of other companies.

The main actors are the users and the customer service staff. The user is directed to FAQ platform, where he can look for his question among a list of frequent questions if he doesn't find it he can use the search bar for further results if still no match is found he then can reach the customer service using the contact form.

The customer service employees are responsible for checking the users' messages and answer their questions and constantly keep the FAQ platform UpToDate.

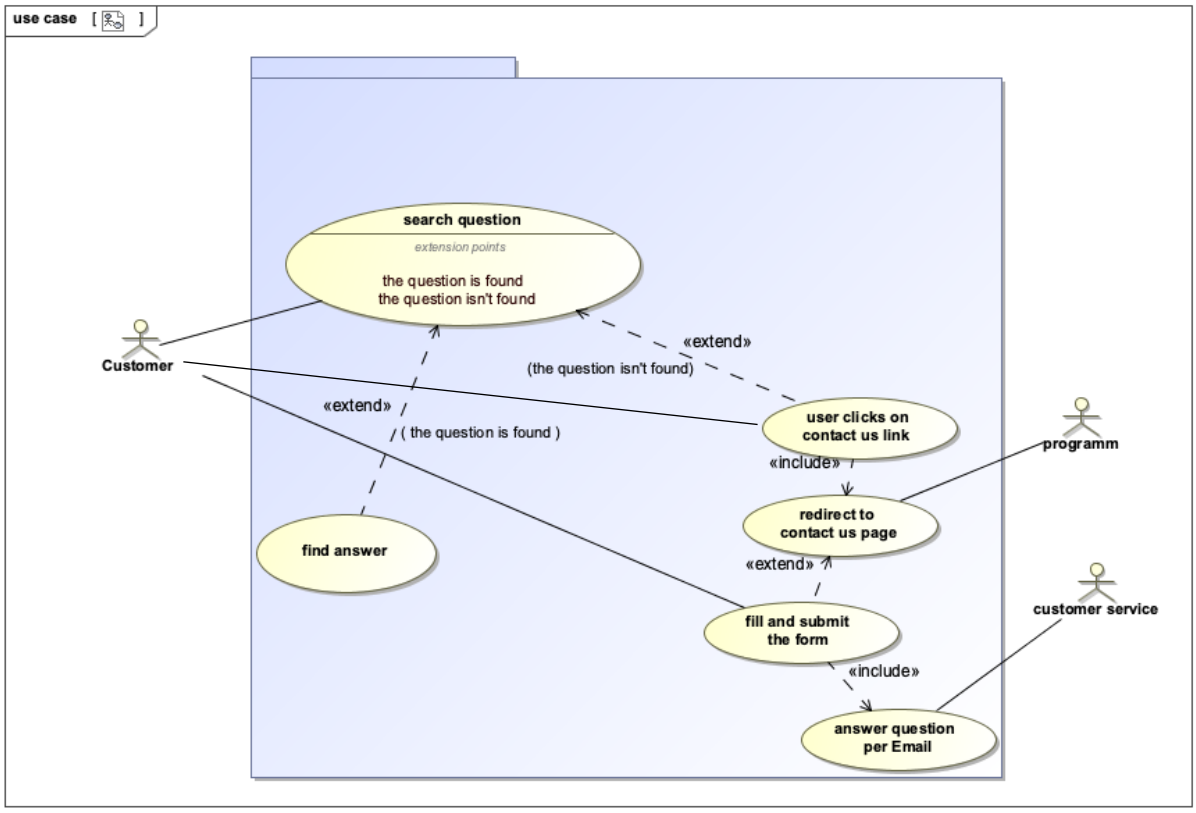


Figure 10.2: FAQ and Contact-us platform, use case diagram

Actor customer

- Customer performs Search question
- Customer performs Clicks on contact us link
- Customer performs Fill and submit the form

Actor Program

- Program performs Redirect to contact us page

Actor Customer Service

- Customer performs Answer question per Email

UseCase Search Question

- Search question is performed by Customer
- Search question is extended by user Clicks on contact us link
- Search question is extended to Find answer

UseCase Find Answer

- Find answer is performed by Customer

UseCase User clicks on contact-us link

- Click on contact-us link is performed by Customer
- Click on contact us link extends Search question
- Click on contact-us link includes Redirect to Contact us page

UseCase Redirect to Contact us form

- Redirect to Contact us form is performed by Program
- Redirect to Contact us form is included by Click on CU link
- Redirect to Contact us form is extended by Fill and submit the form

UseCase Fill and submit the form

- Fill and submit the form is performed by Customer
- Fill and submit the form extends Redirect to contact-us form
- Fill and submit the form includes Answer question per Email

UseCase Answer question per email

- Answer question per Email is performed by Customer Service
- Answer question per Email is included by Fill and submit the form

Class Diagram:

A class diagram doesn't need to be developed here, cause the actors are mainly affected.

Activity Diagram:

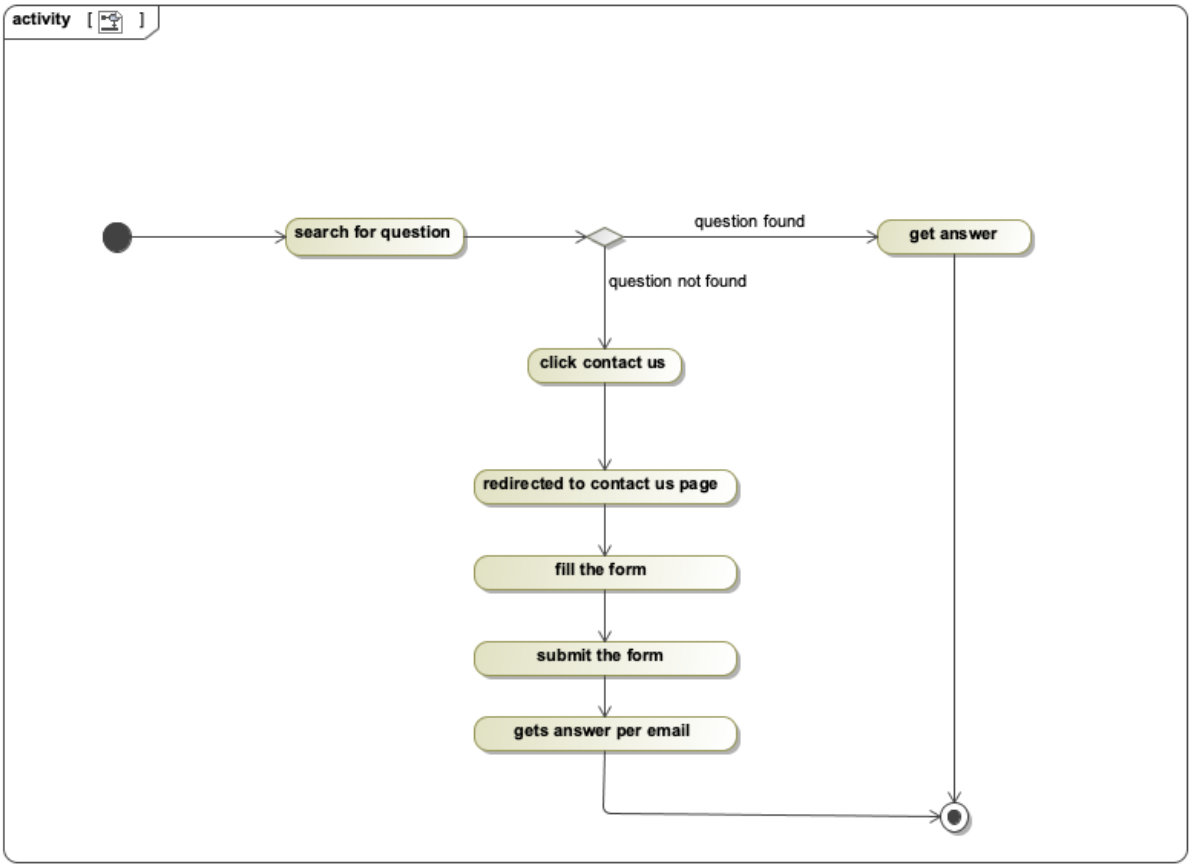


Figure 10.3: Activity diagram of a user searching for his question and contacting CS

Sequence Diagram :

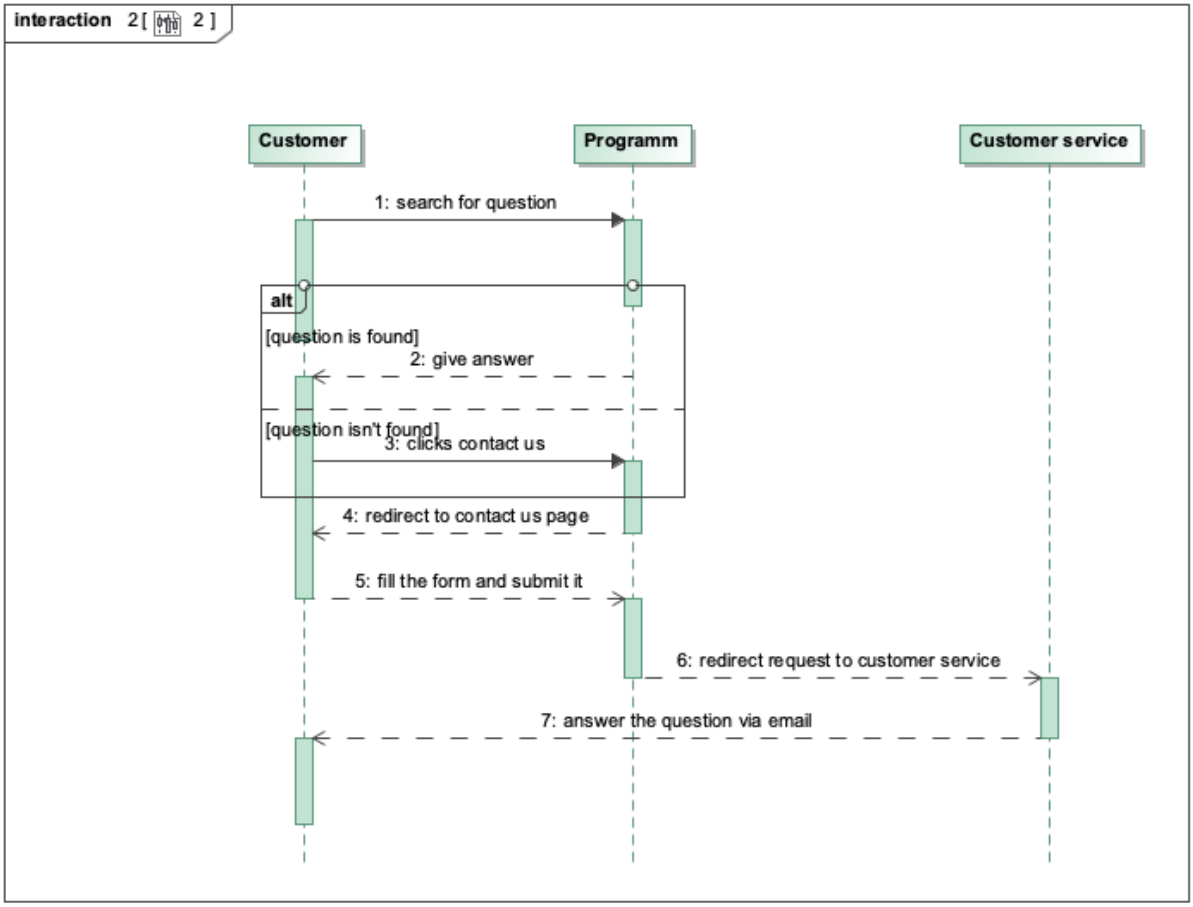


Figure 10.4: Sequence Diagram of user choosing question on FAQ platform

RS- Reward System:

Customer: Orders from the App to earn points After this he can get different rewards (e.g. free delivery after 7 orders, discounts)	after ordering from the App, the customer receives his points and have the choice to use them or keep collecting them to get bigger discounts.

Categorization of the requirement (functional / non-functional):

Functional:

- Order from the App

non-functional:

- The server able to collect the points
- The server able to calculate the suitable discount for each customer

Business value of the corresponding functionality:

This functionality is very important for our business, and it has a high value, which is represented in encouraging our customer to order from our app more and more and also keep up with our concurrent.

Effort Estimation for the implementation of the requirement:

We need to program a software that can calculate the discounts and deliver the right amount to our customer.
The estimated time: 3days-4days.

For UI related functions: UI prototype

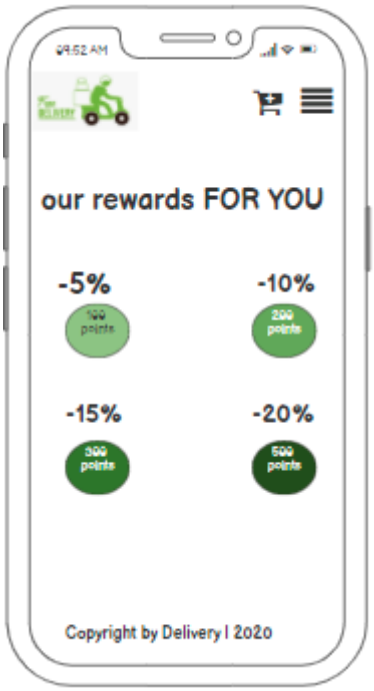


Figure 11.1: *Mobile prototype reward system*

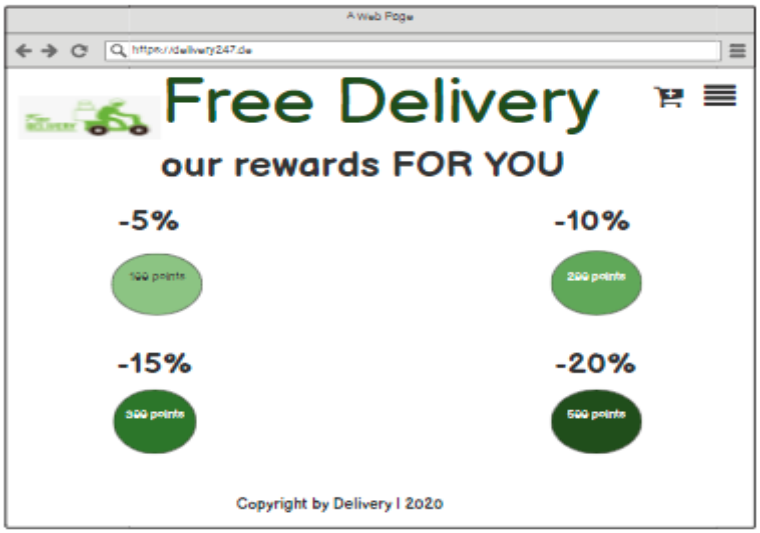


Figure 11.2: *Web prototype reward system*

UML Diagrams

Use Case Scope:

our reward system is implemented to encourage our customer to be loyal to our app, and use it more often thanks to our attractive rewards that cannot be found at our competitor's. for each order there is 5 point to win, if you have reached a certain points, then you get a discount starting with -5% on your next order, u can use it directly or you can choose to keep collecting points for a bigger discount, and that's not the only thing, our customer gets a free delivery after 7 orders.

Use case Diagram:

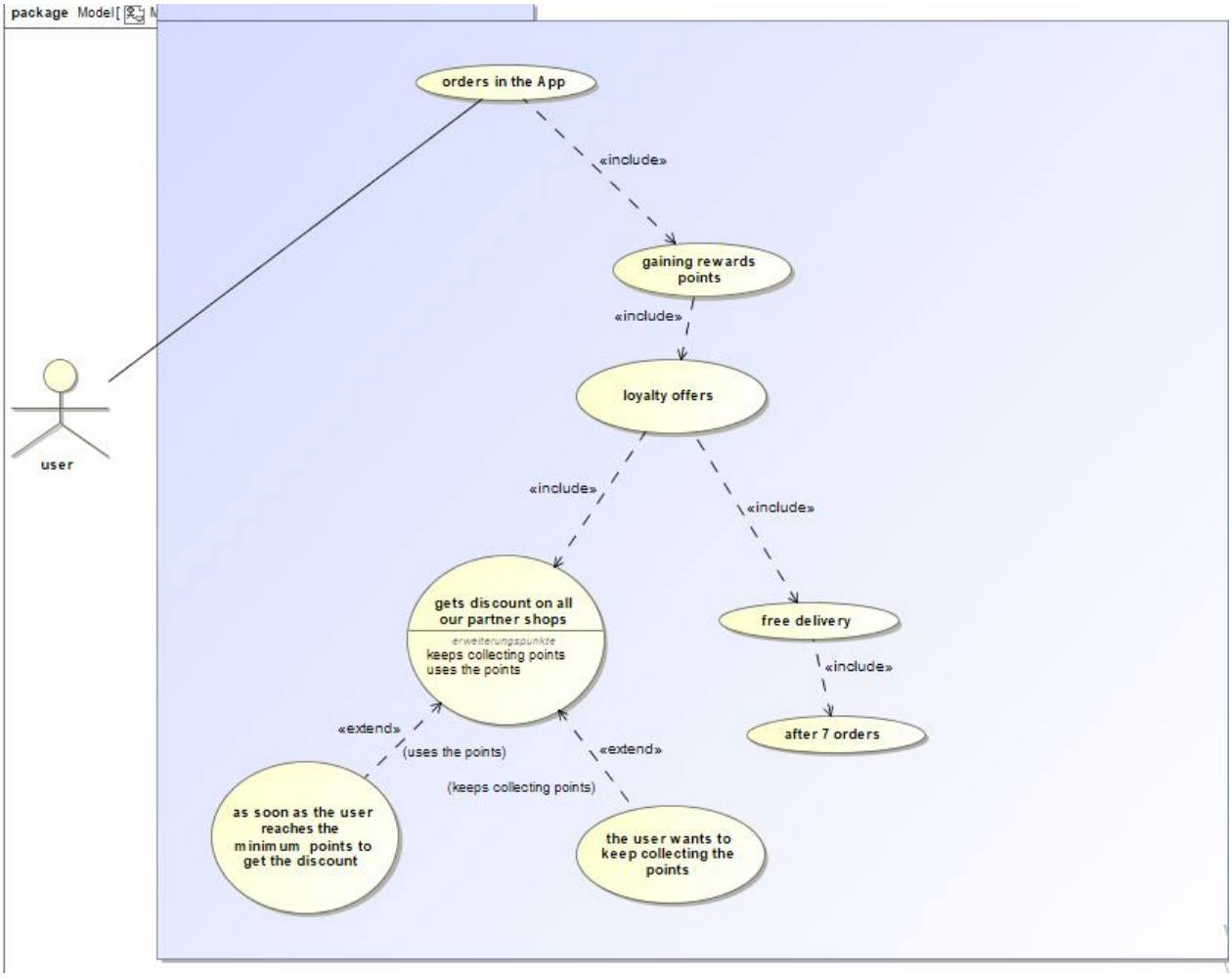


Figure 11.3: *Reward system, use case diagram*

Actor customer:

- Customer performs order in the App
- Customer performs gets the discount
- Customer performs keep collecting the points

Usecase order in the App:

- Order in the App performs by the customer
- Order in the App includes gaining points

Usecase gaining rewards :

- Gaining rewards includes loyalty offers

Usecase loyalty offers :

- Loyalty offers includes gets discount on all our partner shops
- Loyalty offers includes free delivery

Use case free delivery:

- Free delivery includes after 7 days

Use case gets discount on all our partner shops

- discount on all our partner shops includes get the discount
- discount on all our partner shops includes keep collecting points

Use case get the discount

- get the discount performed by the customer

Usecase keep collecting points

- keep collecting points performed by the customer

Class Diagram:

The class Diagram will not be needed in our case, and that's because we only have one actor.

Activity Diagram:

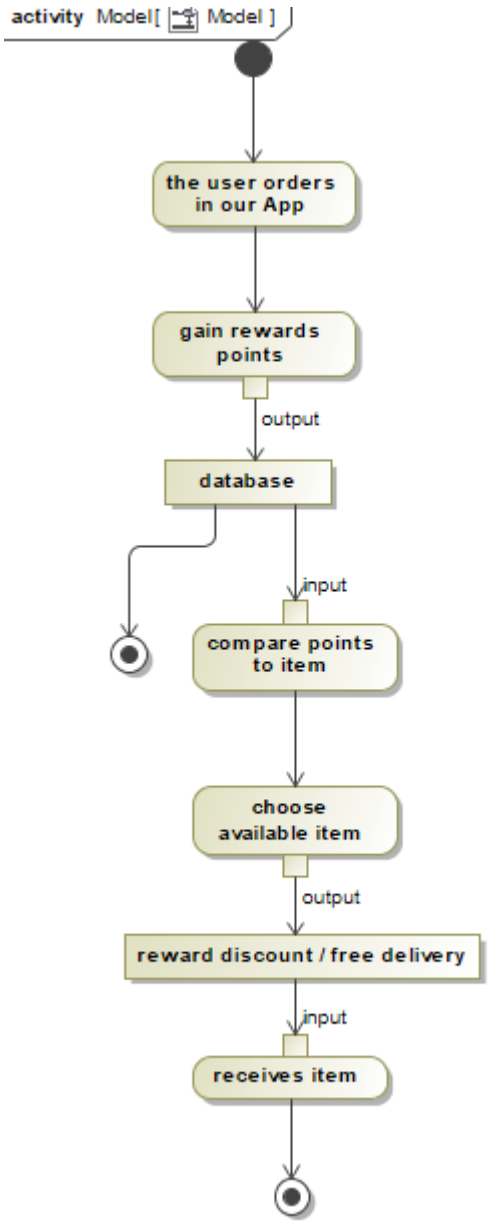


Figure 11.4: *Activity Diagram*

Sequence Diagram:

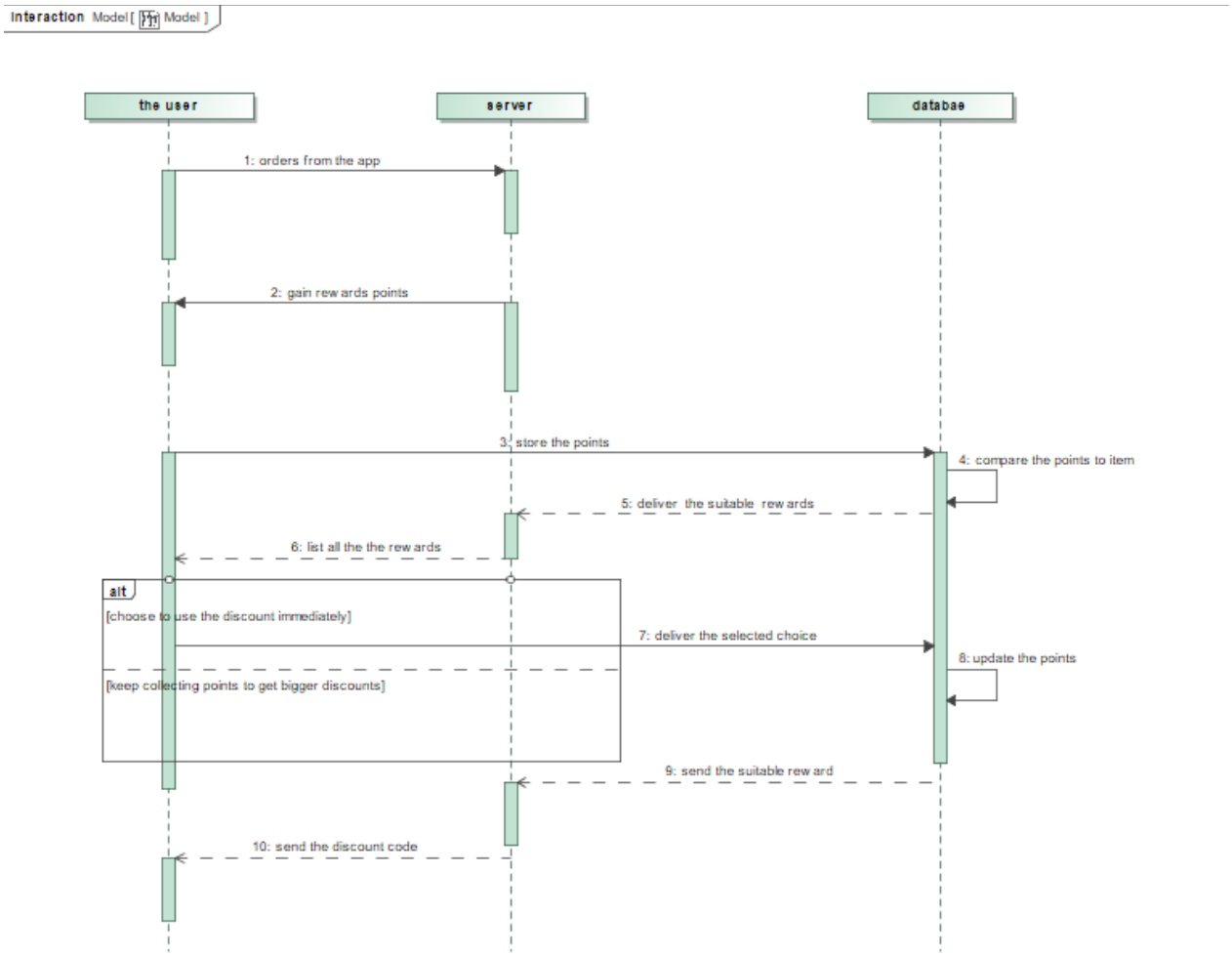


Figure 11.5: Reward System , sequence diagram

SAA-Social media and Advertising:

Company: Increases the Number of the customer by advertising under the condition that the revenue should be higher than the costs of advertising.	Reaching people through our Ads, and being known

Categorization of the requirement (functional / non-functional)

Functional:

- the customer seeing our Ads while scrolling in the internet

Non-functional:

- our server is able to send ads whenever a person searches for a delivery App on the Internet

Business value of the corresponding functionality

Advertising has a high business value, through Ads people can discover our App
And start using it.

Effort Estimation for the implementation of the requirement

We need a graphic designer to make an advertising poster, and a programmer to program a good algorithm.
Estimated time: 7 days.

UML Diagrams

Use Case Scope

The advertising helps our company to grow and gain a popularity, but before investing in the advertising we should have a budget for that, therefor we have to control our profit and then decide if the advertising can be done or not, and which percent can be invested in each platform.

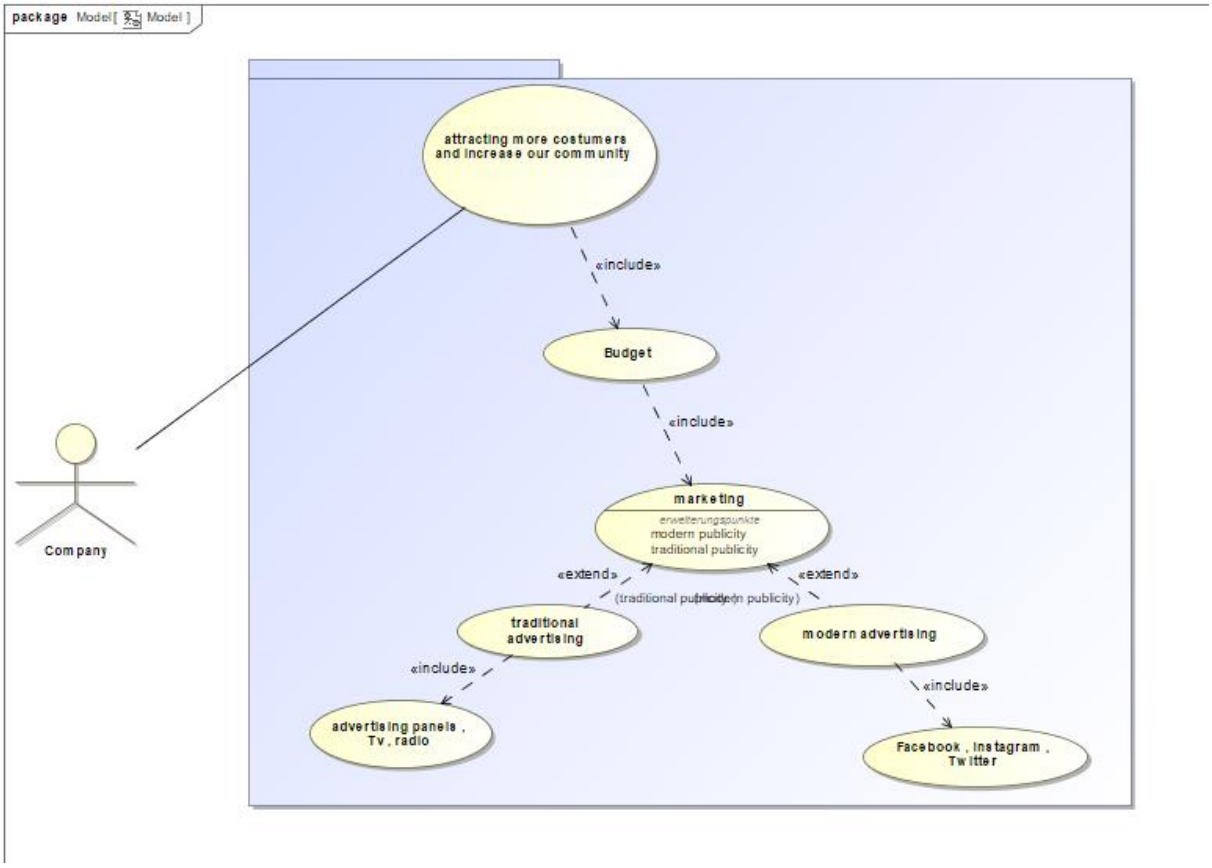


Figure 11.6: Advertising, use case diagram

Actor Company:

- Company performs attracting customer
- Company performs verify the budget
- Company performs marketing

Use case attracting customers:

- attracting customers performed by the company
- attracting customer includes verifying the budget

Use case verify the budget

- verify the budget is performed by the company

- verify the budget includes marketing

Use case marketing

- marketing is performed by the company
- marketing includes modern advertising
- marketing includes traditional advertising

Use case modern advertising

- modern advertising includes Facebook, Instagram, Twitter

Use case traditional advertising

- Traditional advertising includes advertising panels, Tv, Radio

Class Diagram:
The class Diagram will not be needed in our case, and that's because we only have one actor.

Activity diagram:

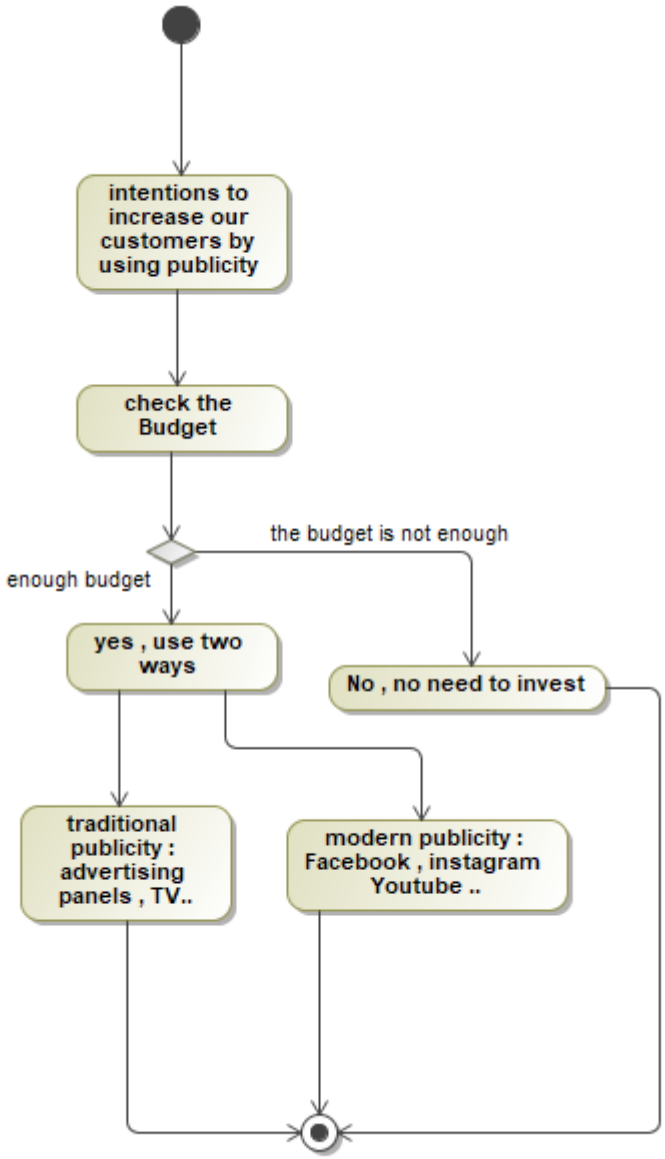


Figure 11.7: Advertising , activity diagram

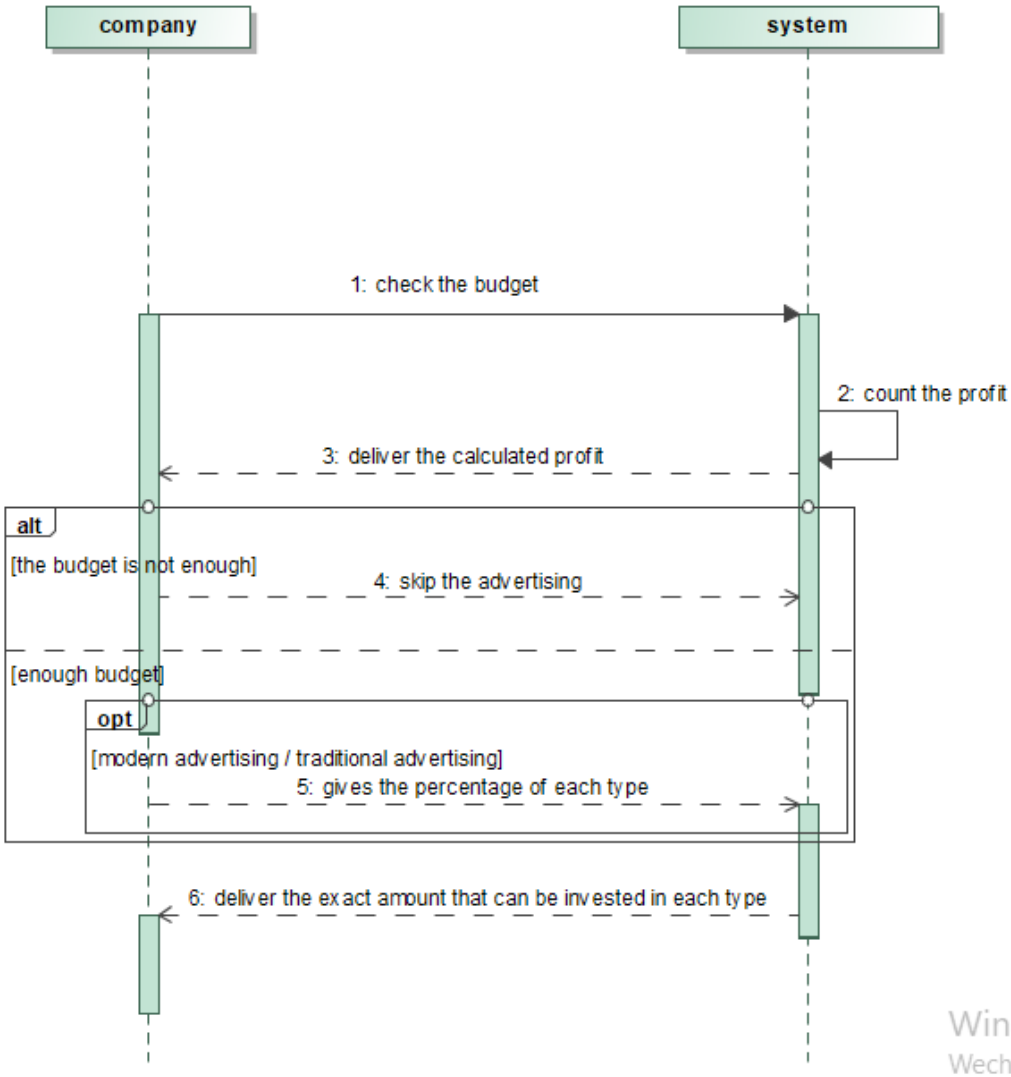


Figure 11.8: Advertising sequence diagram

Retrospective #3
After our meeting we chose to split the retrospective into two parts: “What went well “ and “things that should be improved “

What went well?

- We easily get to these three points.
- Least Discussions and Much of teamwork.
- Ideas are simple to be shown in our backlog.

What need to be improved?
- We are not really innovative here.
- We haven’t gone deep in the revenue of our company.
- A lot of works are still in process.