CarFinder –
System Specification

Project Name	Carfinder
Project Manager	Leon Kuchinka, Leon Schlömmer, Lukas Stransky
Document Owner	Requirements Engineer
Created on	16/11/2017
Last modified	3/2/2018
State	finished
Dokumentablage	SystemSpecificationTemplate.doc

Revision Nr.	Date	Version	Chapter	Modification	Author
1	16.11.17	0.1	2	Functional Requirements Intro, Use case Diagram	All
2	21.11.17	0.2	2	Introduction for every use case	All
3	23.11.17	0.3	2	Characteristic Information for every use case	All
4	28.11.17	0.4	2	Characteristic Information for every use case	All
5	30.11.17	0.5	2	Open points, Scenario for the standard use / Non-standard use	All
6	5.12.17	0.6	2	Activity Diagrams	All
7	7.12.17	0.7	2	Activity Diagrams	All
8	12.12.17	0.8	1	Class Diagrams	All
9	19.12.17	0.9	2	GUI Designs	All
10	21.12.17	1.0	2	GUI Designs	All
11	11.1.18	1.1	1	Initial Situation / Application Domain / Model of the Application Domain	All
12	18.1.18	1.2	1	Initial Situation / Application Domain / Model of the Application Domain	All
13	23.1.18	1.3	1	Glossary / Goal Definition	All
14	25.1.18	1.4	4/5	Quantity Structure / System Architecture and Interfaces	All
15	30.1.18	1.5	6	Acceptance Criteria	All
16	1.2.18	1.6	3	Non-functional Requirements	All
17	3.2.18	1.7	none	Overall layout, last check	All

Content

1 Initial Situation and Goal

1.1 Initial Situation

The currently existing car finding solutions are often too overcomplicated. Our system will have great usability, even for non-car-guys and non-tech-nerds, which means that the home page should have just few buttons, like a big “Find your next car” button, and a smaller login button. Further, the already existing car-finders also implement a used car market place, which we think is recursive, because plenty such services already exist. So, in order simplify the user experience, advertisements will link the user to such websites.

As described in the project proposal, we think there is an appropriately sized marked for our project, since there are no similar solutions in the Austrian market.

Also, the effort needed to create a service like ours is rather manageable, and in case it’s being picked up well by our customers, it’s well scalable.

1.2 Application Domain

One of the problems that we want to solve is, as already mentioned in the project proposal that many Austrians own cars, which they don’t like. So, we want to simplify the process of choosing the perfect car. We want to give every Austrian the chance, to find the car, which they always desired, by asking them a few simple questions.
The sum of the asked questions and corresponding answers is called a search query.
A user can also save a search query to their account, to access it at a later point in time. Furthermore, search queries will also be stored for research purposes, for example: which cars do 24-year-old males like most?
These researches could be published and / or sold to car manufacturers to have a better overview of the market, and perhaps improve the car-finding-algorithm when information about the user is present.

1.3 Glossary

<search query> <Collection of necessary questions for finding cars and the corresponding answers.>

<Car dealer / Car salesman> <Professional who wants to sell cars faster by placing ads on our Site.>

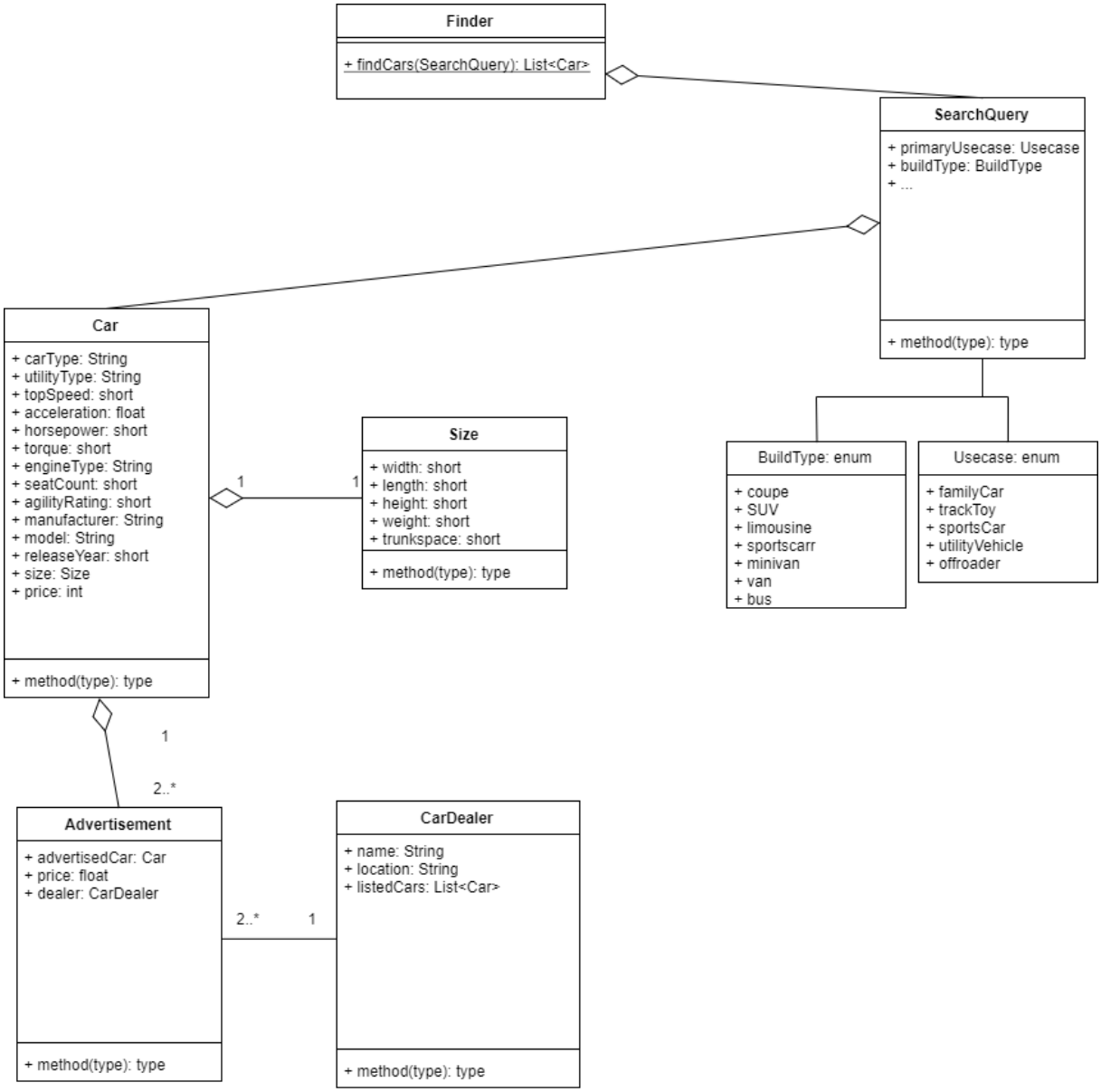
<Primary use case> <Describes a car’s purpose.>

<Build type> <Defines a car’s shape and form>

<Engine type> <Information about the number of cylinders of an engine, and whether it’s Turbocharged, Supercharged or naturally aspirated.>

<Agility rating> <A benchmark for how well a car handles, of how the car feels to drive, handles and performs around tracks>

1.4 Model of the Application Domain



The “Finder” class is the main class to handle every interaction with the user. After the user answered all necessary questions a search query is created and sent to the server, which then returns cars that suit the request. The “Car” class contains all the information that is used to match it to a search query.

When interacting with a new Car Dealer, a new Instance of the “Car Dealer” class is created and added to the database of known Car Salesmen. Furthermore, a dealer can post as many ads as are included in our contract with him.

An Advertisement is shown with a car.

1.5 Goal Definition

Our goal is to give every Austrian the chance to find the car he/she always wanted. We want to finish the project as successful as possible, because we think it could easily help people, who aren´t into cars, still find a car they are happy with. This means the user won't have to have any knowledge to use our service. We want to build an easily understandable and self-explaining website, so that also non-technical people won't have any problem to use our website. This means our target group is every Austrian, who can use a computer, has access to internet and is not fully happy with his/her current car.

The final project will be published in form of a website on a webserver, because web-sites is the most convenient option for a service most people will only use a few times, maybe just once.

2 Functional Requirements

2.1 Client / Server tasks

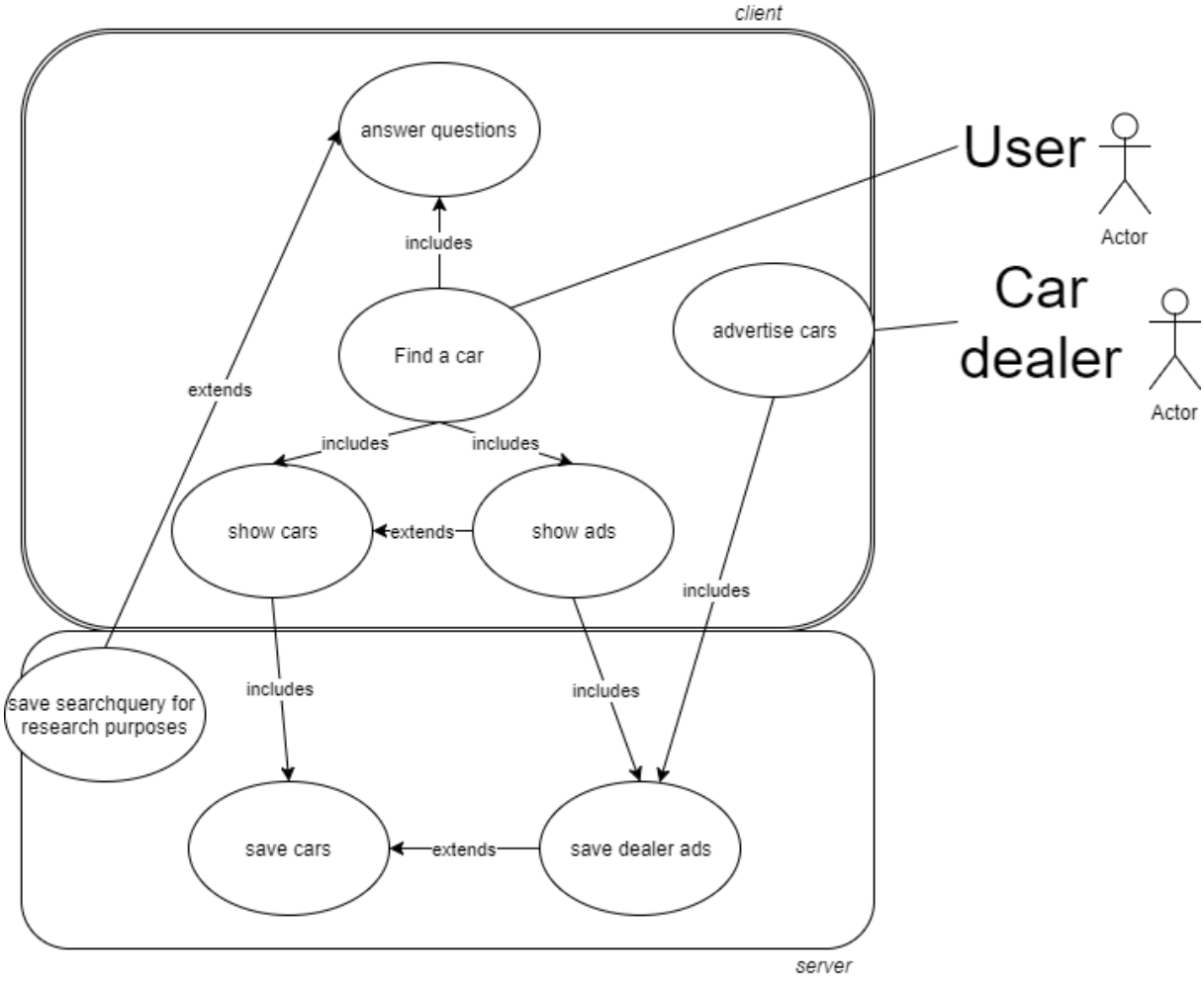
Client:

- Find a car:
  - Central use case for users to find a car that perfectly suits them
- Answer questions:
  - The user uses the program to find a car by answering questions
  - Is included in “Find a car”
- Save search query
- Show cars:
  - Show the cars found by “Find a car”
  - Is included in “Find a car”
- Advertise cars:
  - Central use case for car dealers to list cars for sale
- Show ads:
  - Show the cars advertised by car dealers which suit the query
  - Is included in “Find a car” and extends “Show cars”

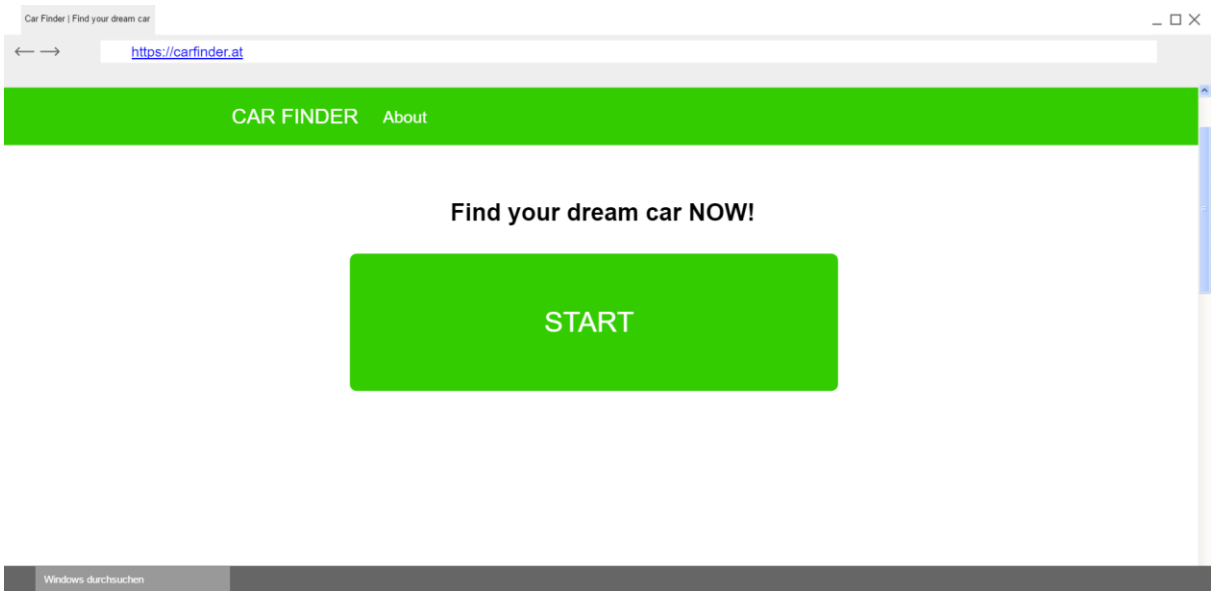
Server

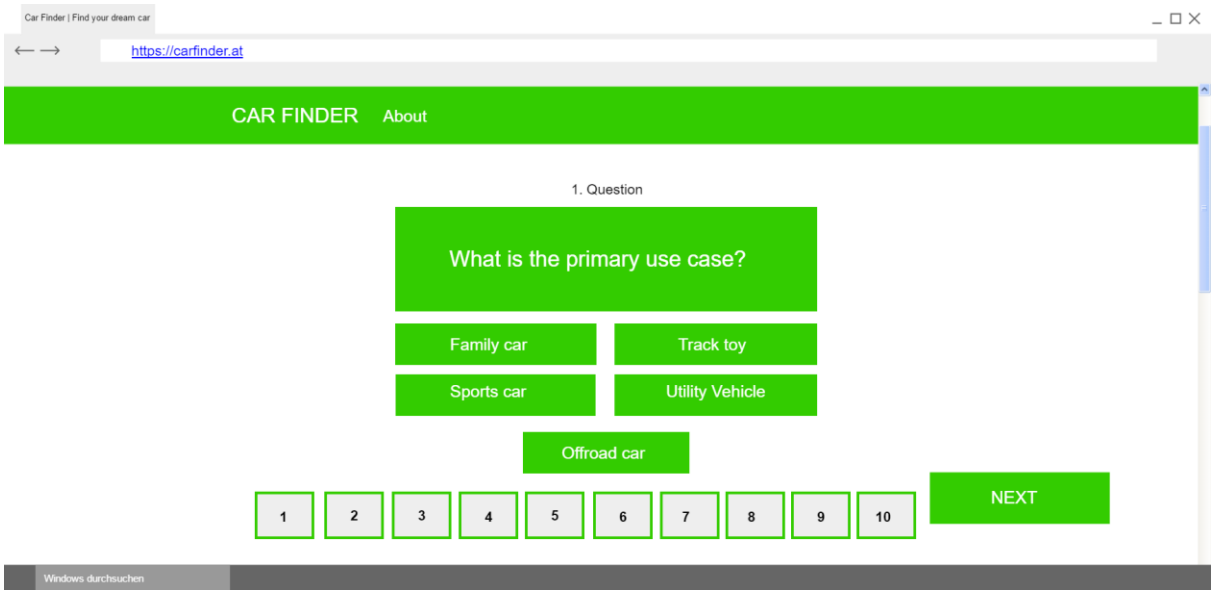
- Save cars
  - To show cars, said cars must be saved
  - Is included in “Show cars”
- Save dealer ads
  - To show dealer ads, said ads must be saved
  - Is included in “Advertise cars”, “Show cars” and extends “Save cars”
- Save search query for research purposes
  - Data should be collected to improve search results
  - Extends “Save search query”

2.2 Use Case Diagram



2.3 General GUI to call the use cases





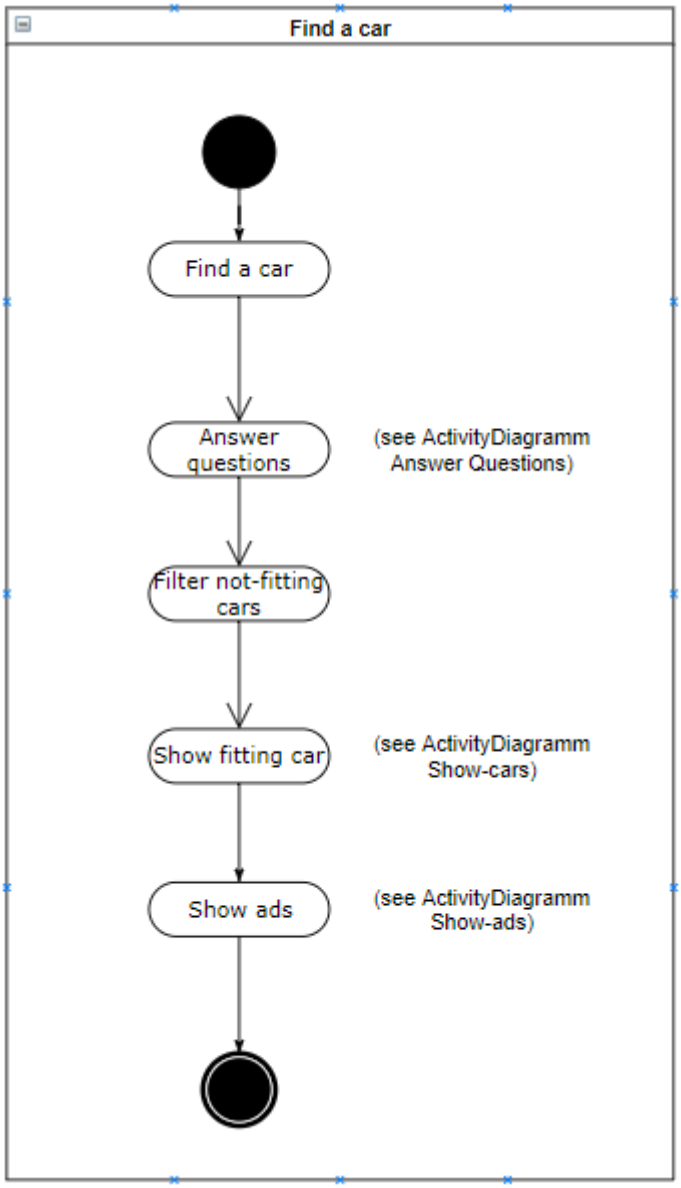
2.4 Use case details 1 – Find a Car

This is the central use case for users to find a car that perfectly suits them, which includes that perfectly fitting cars as well as suiting ads will be shown.

2.4.1 Characteristic Information

Superior business process:	The user opens the website and starts the Car-finding-process.
Goal:	Finding car/cars that suits the input of the user best.
Precondition:	The user wants to find a suiting car for him.
Postcondition:	Suiting cars and ads will be shown.
Involved User:	Only the user that starts the process
Triggering Event:	The user wants to find a suiting car.

2.4.2 Activity Diagram



2.4.3 GUI to call the use case

See 2.2 General GUI to call the use cases

2.4.4 Scenario for the standard use

Step	User	Activity
1	Standard User	Call website
2	Standard User	Click start-button
3	Standard User	Make needed inputs
4	Application	Show cars
5	Standard User	Leave website

2.4.5 Open points

Internet

- WIFI must be activated
- Address of website could be incorrect
- No WIFI signal

2.4.6 GUI for the standard use

See 2.2 General GUI to call the use cases

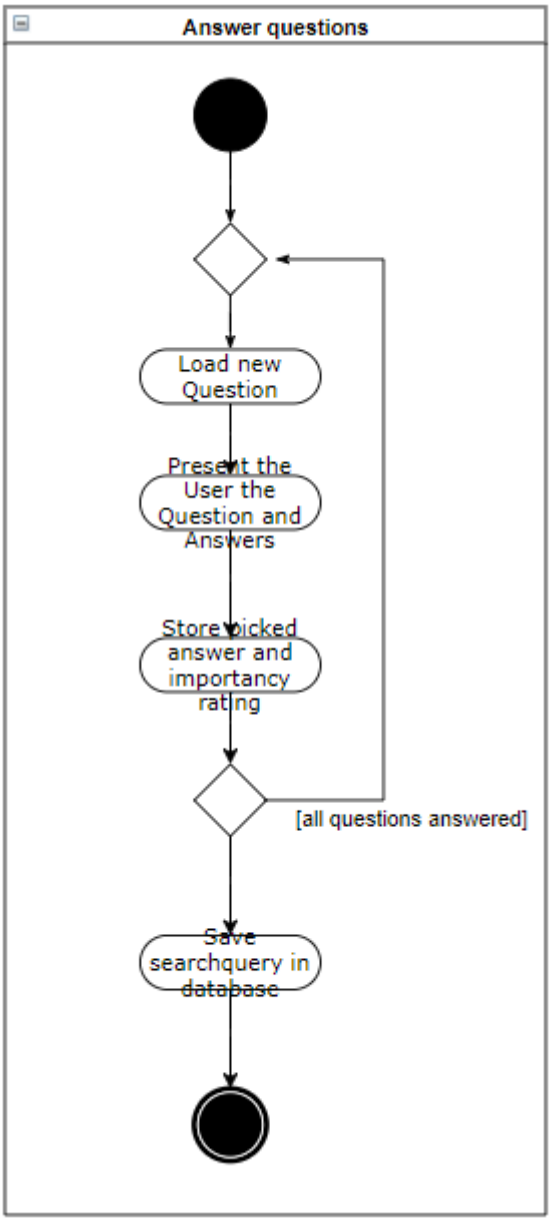
2.5 Use case detail 2 – Answer questions

This use case is included in “Find a car”. The user must answer questions and give them an importance-factor.

2.5.1 Characteristic Information

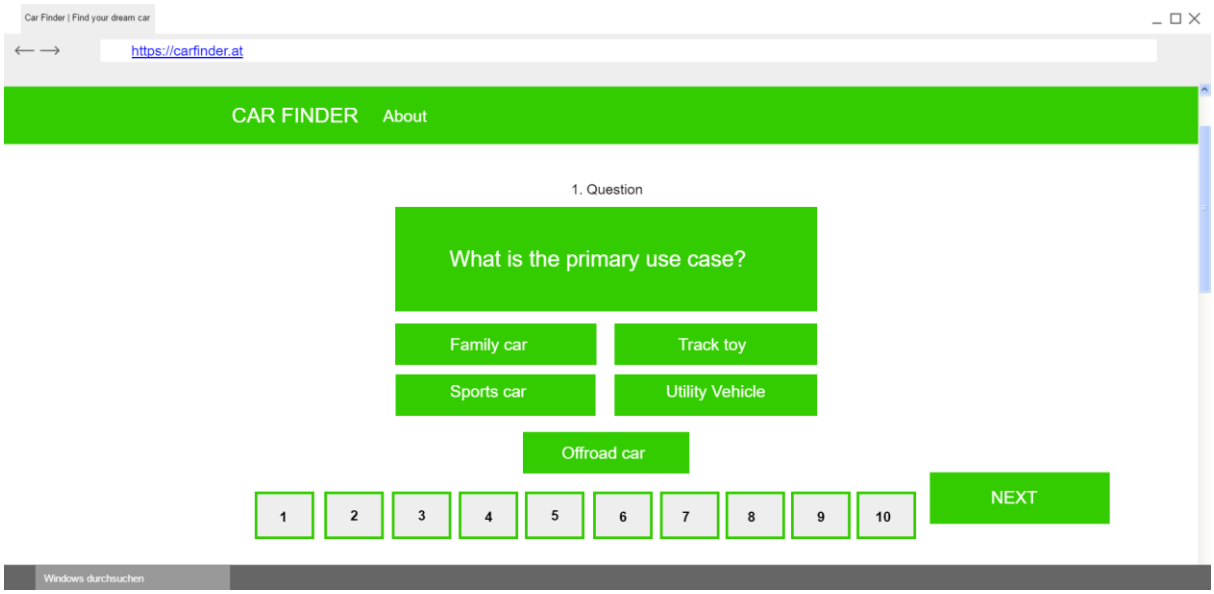
Superior business process:	The user starts the “Find a car”-use case.
Goal:	Getting needed information by receiving inputs.
Precondition:	The user starts the “Find a car” -use case.
Postcondition:	Needed information is ready to be used in the finding-car-algorithm.
Involved User:	Only the user that starts the car finder.
Triggering Event:	“Find a car”-use case is started.

2.5.2 Activity Diagram



2.5.3 GUI to call the use case

See 2.2 General GUI to call the use cases



2.5.4 Scenario for the standard use

Step	User	Activity
1	Standard User	Call website
2	Standard User	Click start-button
3	Standard User	Pick an answer
4	Application	Give the answer an importance-factor
5	Standard User	Click next
6	Standard User	Show cars
7	Standard User	Leave website

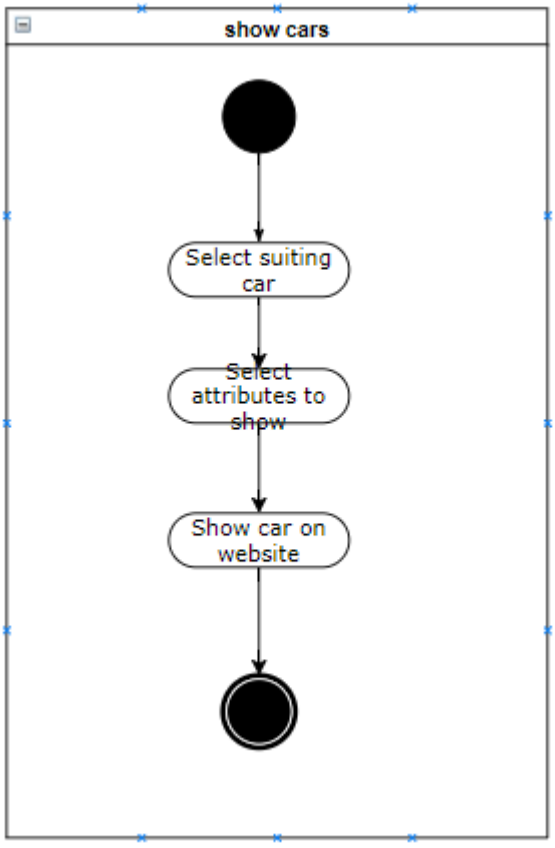
2.6 Use case detail 3 – Show cars

Suiting cars are shown after the “Find a car”-algorithm is finished.

2.6.1 Characteristic Information

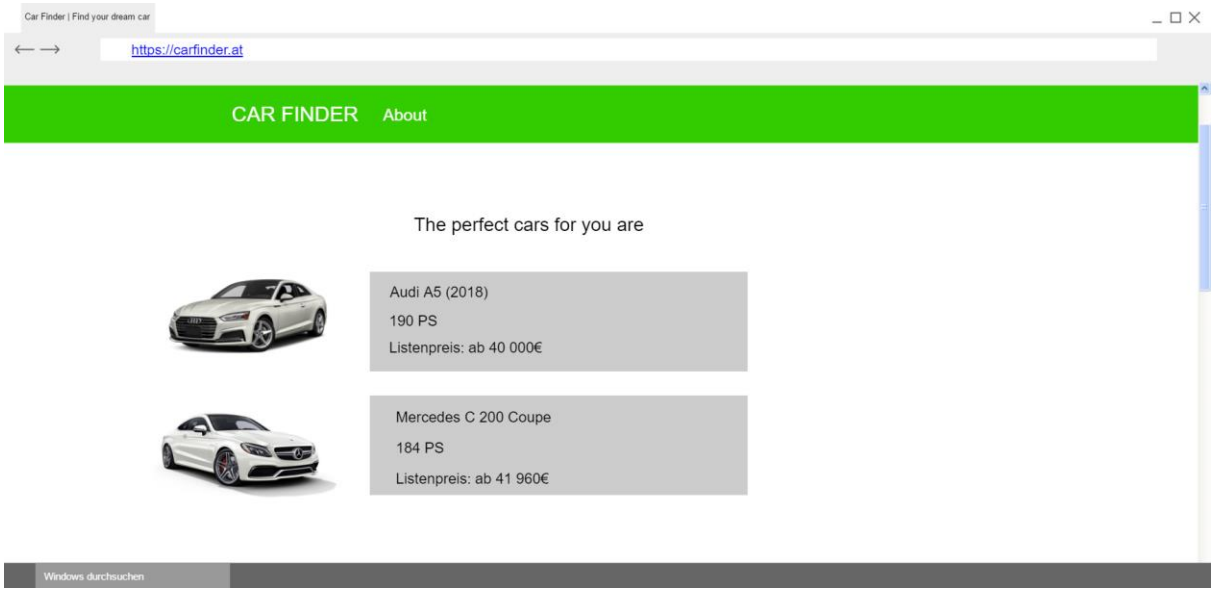
Superior business process:	The user finished “Find a car”-process which includes having all questions answered.
Goal:	Show the cars that fit the input of the user perfectly.
Precondition:	The user answered all questions.
Postcondition:	User can scroll through his top-matching cars.
Involved User:	Only the user.
Triggering Event:	“Answer questions”-use case is finished.

2.6.2 Activity Diagram



2.6.3 GUI to call the use case

See 2.2 General GUI to call the use cases



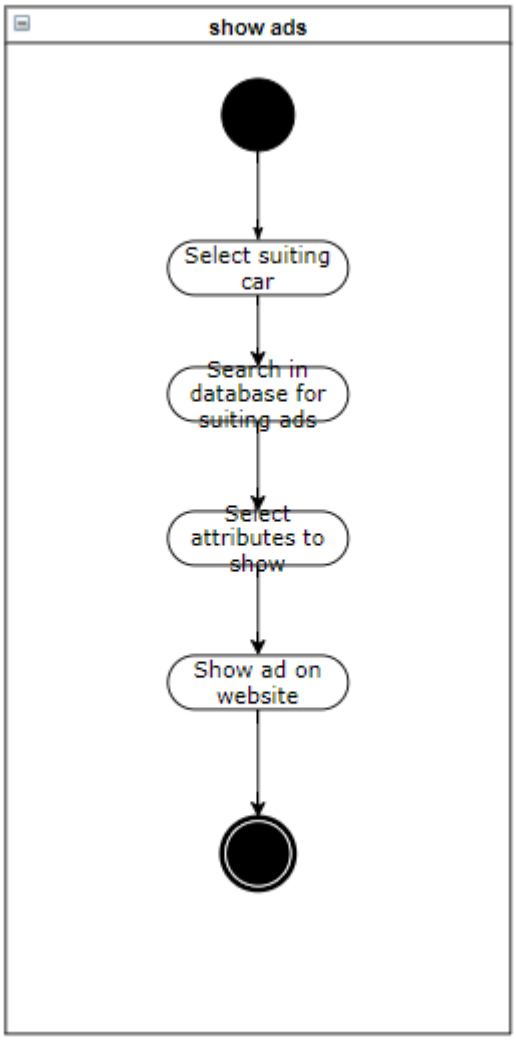
2.7 Use case detail 4 – Show ads

Suiting car-ads which are advertised by car-dealers are shown after the “Find a car”-algorithm is finished.

2.7.1 Characteristic Information

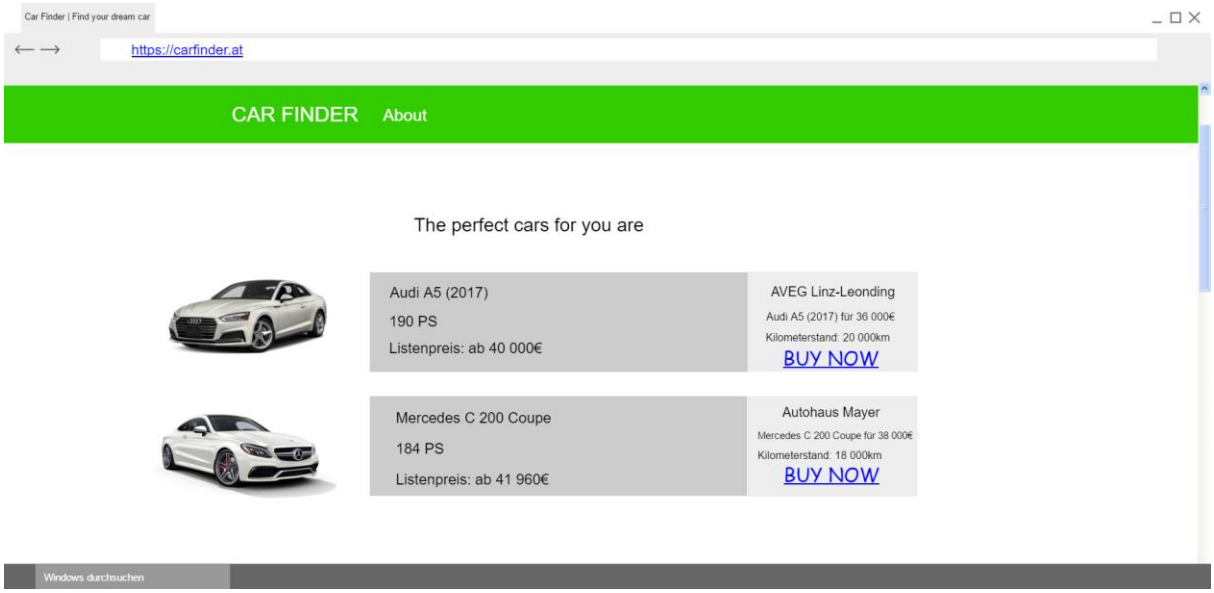
Superior business process:	The user finished “Find a car”-process which includes having all questions answered.
Goal:	Show the cars advertised by car dealers which suit the query.
Precondition:	The user finished the “Find a car”-use case.
Postcondition:	User can scroll through offers of car dealers.
Involved User:	Car dealer, User
Triggering Event:	“Find a car”-use case is finished.

2.7.2 Activity Diagram



2.7.3 GUI to call the use case

See 2.2 General GUI to call the use cases



2.7.4 Open points

Advertiser

- There are no offers for a found car from any advertiser

2.8 Use case detail 5 – Advertise cars

This is the central use case for car dealers to list cars for sale.

2.8.1 Characteristic Information

Superior business process:	Car-dealer wants his car-offers to be shown on the website.
Goal:	Adds are stored in the database and are shown if they suit to a search query.
Precondition:	Car-dealer wants his car-offers to be shown on the website.
Postcondition:	Car-offers are stored into the database and are shown on the website.
Involved User:	Car dealer
Triggering Event:	Car-dealer wants his car-offers to be shown on the website.

2.8.2 GUI to call the use case

See 2.2 General GUI to call the use cases
See 5.2 GUI to call the use case

2.8.3 Scenario for the standard use

Step	User	Activity
1	Car dealer	Apply to take his ads
2	Developer	Check if ad is interesting
3	Developer	Decide whether to take it or not
5	Developer	Decide to take it
6	Car dealer	Provide us with all the information needed
7	Developer	Save information’s into the database

2.8.4 Scenario for the non-standard use

Step	User	Activity
1	Car dealer	Apply to take his ads
2	Developer	Check if ad is interesting
3	Developer	Decide whether to take it or not
4	Developer	Decide not to take it

2.8.5 Open points

Advertiser

- There are no advertisers
- We cannot afford advertisers
- Advertisers are not interested
- We cannot convince some advertisers

2.9 Use case detail 6 – Save dealer ads

Car-dealer’s advertisements are stored in the database.

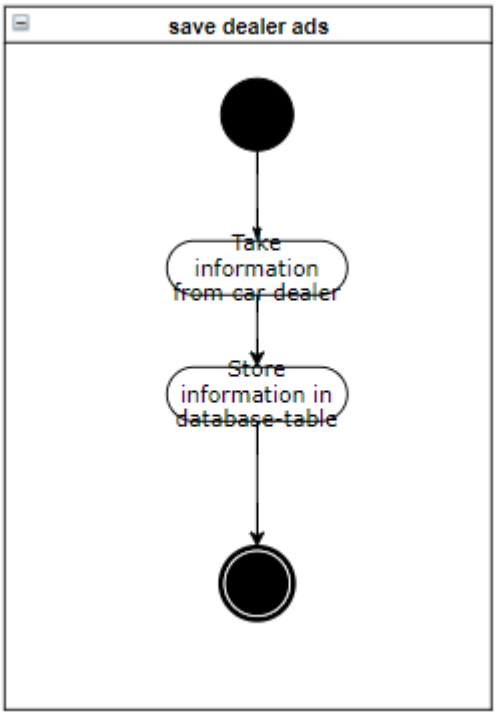
2.9.1 Characteristic Information

Superior business process:	Car-dealer wants to advertise cars, which includes saving the ad in a database.
Goal:	An advertisement is stored in the database and ready to be accessed by the “Show ads” use case.
Precondition:	Car-dealer has filled out the details to a specific car and wants an ad to be shown.
Postcondition:	A new entry in the ad-database is made.
Involved User:	Car-dealer
Triggering Event:	Car-dealer wants to show an ad.

2.9.2 GUI to call a use case

No GUI needed

2.9.3 Activity Diagram



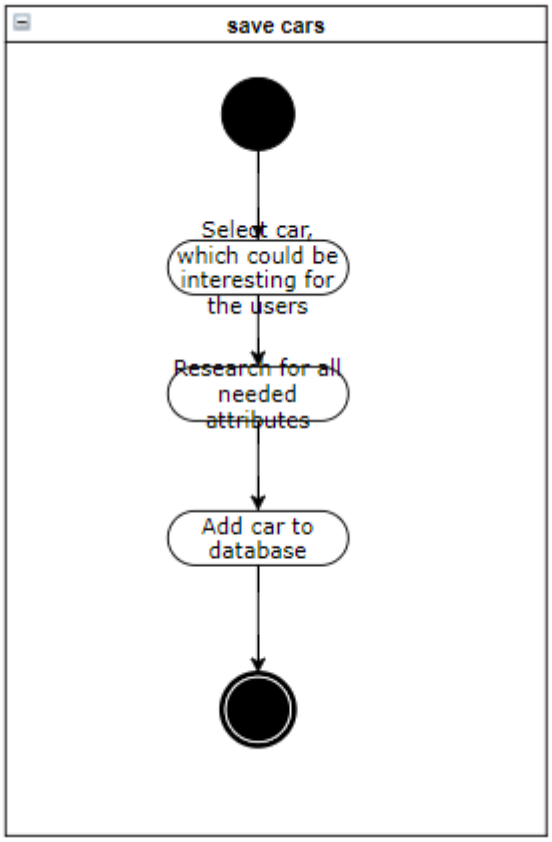
2.10 Use case detail 7 – Save cars

The main function of this use case is to store new car data into the database.

2.10.1 Characteristic Information

Superior business process:	Developer wants or needs new car data to be stored into the database for the use case “Find car” and “Show Car”.
Goal:	Car data is stored into the database and is ready to be accessed by the “Show Car” and “Find Car” use case.
Precondition:	Developer wants or needs new car data to be stored into the database for the use case “Find car” and “Show Car”.
Postcondition:	Car data is stored in the database and are ready to be accessed.
Involved User:	Developer
Triggering Event:	Developer wants or needs new car data to be stored into the database for the use case “Find car” and “Show Car”.

2.10.2 Activity Diagram



2.10.3 GUI to call the use case

No specific GUI needed

2.10.4 Open Points

Car data

- We have no right to use this specific car data
- The Storage space of the database is empty

2.11 Use case detail 9 – Save search query for research purposes

A search query made by a user is stored automatically, to be accessed later for statistics, etc.

2.11.1 Characteristic Information

Superior business process:	The search query is completed.
Goal:	Store the answers to the questions for statistics and researches.
Precondition:	The user answered all questions.
Postcondition:	Search query is saved in a database.
Involved User:	Application, User
Triggering Event:	All questions to find a car are answered and saved to the query.

2.11.2 GUI to call the use case

No GUI is needed

3 Non-functional Requirements

3.1 Type USE: Usability requirement

To use the project “Car Finder”, some important criteria must be satisfied:

- The website’s loading time and the server response time should be at least under 5 seconds, because nobody would use a website, which takes longer to load than that.
- The time to find the right car should not use more than 7 minutes, because attention would be lost if it takes too long.
- The appearance is also very important, because websites with fixed elements aren´t attractive so the website should be dynamically resizable.

3.2 Type EFFIC: Efficiency requirement

As described in “Type USE”, the loading time should not be longer than 5 seconds. Therefore, we must optimize the speed of reading the data from the database as well as the car-finding-algorithm itself. Now, we are not able to calculate loading times.
Since our project is a website, our users won´t need memory for it, accept the one for cookies but this is negligible.

3.3 Type MAINT: Maintenance and portability requirements

So far, there are no changes in sight. There is a possibility of adding other languages like English at a later point in time.

3.4 Type SEC: Security requirement

Since we don’t safe user’s data, except the search query, the data security is not a big issue in our project. Nevertheless, our car data and the stored search queries are not available to normal users. Only our team can access it.

Talking about availability, our website should be accessible all the time.

3.5 Type LEGAL: Legal requirement

Data and pictures of the cars will only be used if we have the right to use them. We do not know other laws or standards, which could be important.

4 Quantity Structure

In this part of the document, the expected number of users, stored cars and dealers will be discussed.

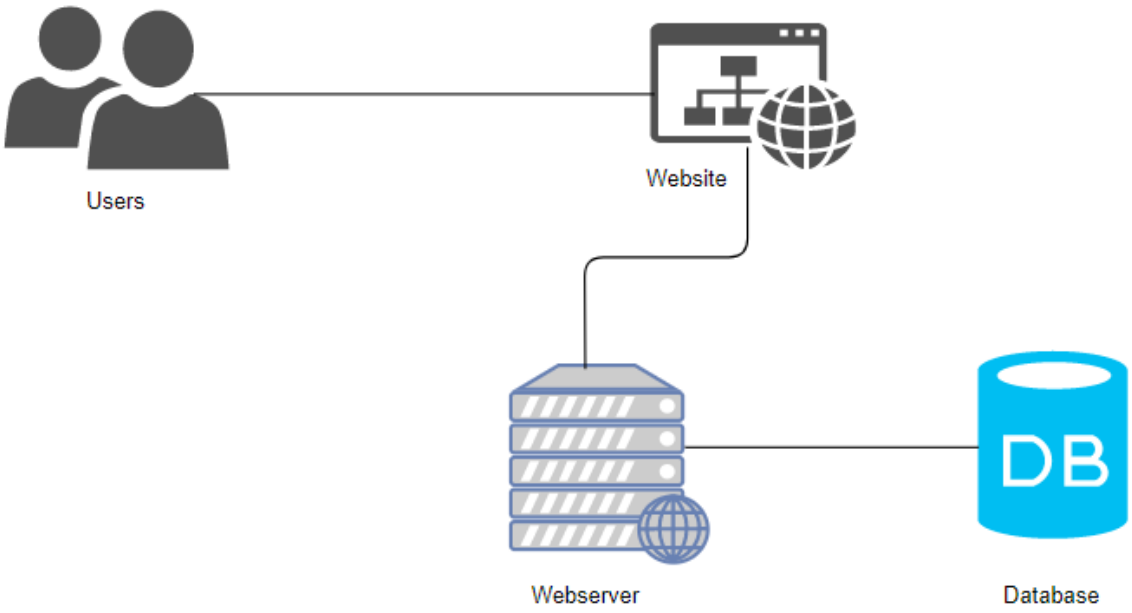
Structure	**Expected Quantity**	**Involved Use Case**
Cars saved in database	100	Find a Car
Users every day	300-400	Find a car
Expected Advertisers	10-100	Advertise Cars
Expected search queries / hour	30	Find a car
Expected visits per day *(don't confuse with users every day, some users might visit twice or often)*	500-700	Find a car

We expect our clientele to be rather small at first. However, when our service is being picked up well and our clientele grows quickly, that is not a problem, because of our projects scalability.

Also, no research must be done in the quantity of questions a user is asked to evaluate the perfect car for them, since these questions are (at least for the time being) rather static and not going to change very often, than dynamic and being updated a lot.

5 System Architecture and Interfaces

The website is the interface to the users, where they get will be able to use our service. The Webserver will host the website and connect to the database, where all the car-data, the questions and answers and the search queries will be stored.



6 Acceptance Criteria

6.1 Use case 1 – Find a car

Test Step	Expected Behaviour
Click start button	First question should be displayed
Find cars	The perfect suiting cars for the user are found by the algorithms

6.2 Use case 2 – Answer questions

Test Step	Expected Behaviour
Choose an answer	Answer button is marked
Assign an importance factor	Importance factor button is marked
Click the next button	The following question should be shown

6.3 Use case 3 – Show cars

Test Step	Expected Behaviour
Finished answering all questions	Show found cars
Scroll through cars	The user can scroll through the found cars

6.4 Use case 4 – Advertise cars

Test Step	Expected Behaviour
Advertiser provides his car data	Data is saved in the database

6.5 Use case 5 – Show ads

Test Step	Expected Behaviour
Finished answering the last question	Matching ads for the found cars are shown
Click on ad	User will be redirected to the advertiser’s website

6.6 Use case 6 – Save cars

Test Step	Expected Behaviour
Found car data	Car data is saved in the database

6.7 Use case 7 – Save ads

Test Step	Expected Behaviour
Advertiser provides his ad	Ad is saved in the database