



Norwegian University of
Science and Technology

Backload: Web application for increased transportation efficiency

Author(s)

Håkon Horvei
Aslak Ihle
Mats Åsbu Jacobsen
Christoffer Seljehaug

Bachelor in Web Development
20 ECTS

Department of Architecture and Design
Norwegian University of Science and Technology,

16.05.2018

Supervisor

Kjell Are Refsvik

Sammendrag av Bacheloroppgaven

Tittel:	Backload: Webapplikasjon for økt transport effektivitet
Dato:	16.05.2018
Deltakere:	Håkon Horvei Aslak Ihle Mats Åsbu Jacobsen Christoffer Seljehaug
Veiledere:	Kjell Are Refsvik
Oppdragsgiver:	Ptt Communications AS, by Johannes Nordsveen
Nøkkelord:	nettside, transport, backload
Antall sider:	67
Antall vedlegg:	17
Tilgjengelighet:	Åpen

Sammendrag:	Å kjøre med ekstra plass på lasteplan er vanlig i transportindustrien. I Norge, så er ca. en tredjedel av den totale mengden kilometer kjørt av transportselskap uten last. Ved å lage en web plattform som kobler sammen transportører med andre firma, så kan vi redusere mengden lastebiler med tomt lasteplan. Dette gjør vi med å fylle opp den ledige plassen med oppdrag som er langs ruten en allerede skal kjøre. Målet er å redusere kostnader for selskap som trenger frakt, og økt profitt for transport selskap, som et biprodukt med reduisering av forurensing. Resultatet av prosjektet er designet til en webløsning og implementasjonen av kjernefunksjonen i en funksjonell prototype.
-------------	---

Summary of Graduate Project

Title:	Backload: Web application for increased transportation efficiency
Date:	16.05.2018
Authors:	Håkon Horvei Aslak Ihle Mats Åsbu Jacobsen Christoffer Seljehaug
Supervisor:	Kjell Are Refsvik
Employer:	Ptt Communications AS, by Johannes Nordsveen
Keywords:	website, transport, backload
Pages:	67
Attachments:	17
Availability:	Open

Abstract:	Driving with extra available space is a common occurrence in the cargo transportation industry. In Norway, approximately one third of the kilometers driven by transportation trucks are without cargo. By creating a web platform that connects transporters directly to other companies, we can reduce the amount of empty space in transportation vehicles. This is done by filling up the available cargo space with delivery requests that are along the route they are already driving. The aim is to reduce overall costs for companies requesting deliveries and increase profit margins for transportation companies, with the added byproduct of reduced pollution. The result of this project is the design of a web application and the implementation of the core functionality in a working prototype.
-----------	--

Preface

We would like to thank Johannes Nordsveen and Rolf Schonhowd for trusting us with this idea and supporting us throughout the project. We would also like to thank our supervisor Kjell Are Refsvik for helping us and giving great inputs along the entire process. Finally we would like to thank everyone who has participated in interviews and user testing. Thank you!

- Håkon Horvei, Aslak Ihle, Mats Åsbu Jacobsen and Christoffer Seljehuag

Contents

Preface	iii
Contents	iv
List of Figures	vii
1 Introduction	1
1.1 Project Description	1
1.2 Goals	2
1.3 Research questions	3
1.4 User groups	3
1.5 Limitations	3
1.6 Report structure	4
1.7 Terminology	4
2 Background	6
2.1 Product owners	6
2.2 Original Idea from Product owners	6
3 Methodology	8
3.1 Brainstorming	8
3.2 Interview	8
3.3 Survey	9
3.4 Qualitative data analysis	9
3.5 Personas and scenarios	9
3.6 Prototyping	10
3.7 Agile development	10
4 Idea development	12
4.1 The Norwegian transportation industry	12
4.2 Competitor research	13
4.3 First iteration	13
4.4 New problems	14
4.5 Survey	14
4.6 Interview	15
4.7 Second iteration	15
5 Design and prototyping	17
5.1 User experience	17
5.2 Features and functionality	17
5.3 Considered functionality	32
5.4 Layout	33

5.5	Final Prototype	37
5.6	User testing	39
6	Implementation	42
6.1	Technologies	42
6.2	Angular	43
6.3	Google Maps	43
6.4	Material	44
6.5	Drag slider	45
6.6	Backend	45
6.7	Developing application	46
6.8	Final result	55
7	Process	59
7.1	Role distribution	59
7.2	Collabaration tools	59
7.3	Contracts	60
7.4	Meeting Times	60
7.5	Logging	61
7.6	Discussion	61
8	Ending	63
8.1	Goals	63
8.2	Future Work	64
8.3	Conclusion	67
	Bibliography	68
A	Meeting logs	72
B	Persona	82
C	Interview and survey questions	86
D	Project plan	88
E	Interview consent form	90
F	Redefining scope	91
G	Competitor research	95
H	Ustertesting consent form	107
I	Prototype testing tasks	108
J	Prototype testing result	109
K	Internal contract	117
L	Gantt chart	119
M	Sketches	120
N	Hour log	121
O	Sass structure	129
P	Code snippets for table	130
P.1	JobListingsDatabase	130

P.2 JobListingsDataSource	131
Q Digital prototype	132

List of Figures

1	Line chart showing the percentage of kilometers driven without cargo for Norwegian transportation trucks from 2008-2017	1
2	Example of a Waze live map search for the route Gjøvik - Molde	19
3	Example of a Bing Maps search for the route Gjøvik - Molde - Trondheim	20
4	Example of a Google Maps search for the route Gjøvik - Molde - Trondheim	20
5	Our initial layout	34
6	Main view split into 3 parts	36
7	Map as background, sidebar and table as overlay. Table hidden.	36
8	Map and table side by side	37
9	The main page of the prototype after date and route input	38
10	Ongoing bidding war on Overview page, input for adding new bid with seperate date	39
11	Our database tables, with all columns and datatypes	46
12	The left side shows the source files for the Angular project. The right side shows the structure for our components	48
13	Example of Google Maps Autocomplete with a list of suggested places	51
14	Visualization of the overview_path property in a route, the white circles represents the geographic points contained in the overview_path array	53
15	Visualization of how distance between route and pickup location is calculated from overview_path	54
16	The initial view of the page	56
17	The route between Oslo and Bergen has been plotted	56
18	The route Oslo - Bergen - Ålesund	57
19	Shows the route Oslo - Bergen - Ålesund where start date and end date has been entered	57
20	Shows how the drag slider has been used to give more room for the table	58
21	Shows an expanded table row	58

1 Introduction

1.1 Project Description

Approximately a third of the total distance traveled by transportation trucks registered in Norway is without any cargo. During the last nine years there has been some fluctuation in the percentage of kilometers driven empty. As the linear trendline in Figure 1 shows, there has been a steady increase of kilometers driven without cargo, demonstrating a definite room for improvement. (Økt transportarbeid)

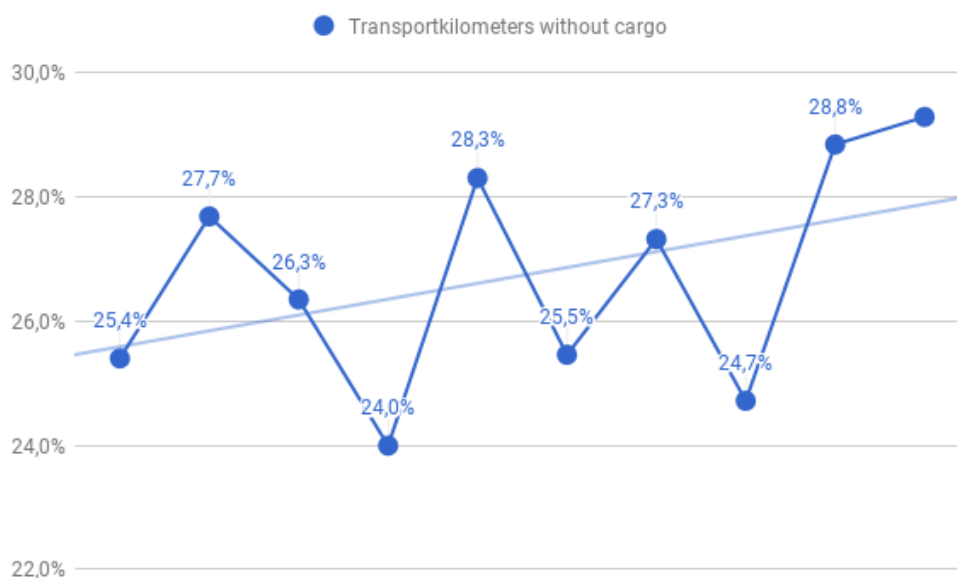


Figure 1: Line chart showing the percentage of kilometers driven without cargo for Norwegian transportation trucks from 2008-2017

In the transportation industry, there is often a third party involved in a transportation assignment. This third party is a broker connecting the two parts and taking a cut for doing the job. Our solution is a web application named Backload that acts as an alternative third party, which for the employer would cost less and require little time to use. The benefit for the transporter would be having more available transportation assignments at any given time.

The end product of our project will be a working prototype of a web application that acts as a marketplace for transporters and companies that want something transported. The thought is that the web application should serve as a complimentary platform for acquiring transportation assignments, it is not intended as a solution to replace the current

methods used to acquire transportation assignments. Backload will be used as a way of finding additional transportation assignments along or nearby pre-planned routes with available cargo space, so the otherwise unused space can be utilized with minimal time loss.

1.2 Goals

The project goals are divided into three categories; development goals, impact goals, and personal goals. Each category define a set of goals for what we intend to achieve with the project. The development goals are set in relation to not just what we want to create, but also how we approach them. The impact goals are set in relation to what beneficial outcomes we want the finished product to bring in to the world.

Defining project goals is not only helpful in the sense that it clarifies the purpose of the project, it also encourages progress towards a desired outcome. This enables us to gain a better understanding in what steps will be required to accomplish our goals.

1.2.1 Development goals

- Create a working prototype for the core solution that both the product owner and the group can be proud of.
- The application should be a fully responsive web application that can be used on the majority of all browsers.
- Make use of a wide range of technologies, frameworks and tools.
- Follow industry-standards and best practices.
- Follow the Scrum methodology.
- Make use of good UI/UX design to create a user interface that appropriately reflects the needs of the target users, while also being easy to understand and use.
- The application should be scalable. It should be possible to add functionality for other target audiences at a later stage.

1.2.2 Impact goals

- Help entrepreneurs find qualified transportation companies.
- Make it easier for businesses to plan more efficient routes.
- Reduce the amount of empty space in transportation vehicles by filling up the cargo space.
- Help reduce pollution by decreasing the amount of trucks needed to move the same amount of cargo.
- Reduced costs for entrepreneurs and companies, and higher profits for transportation companies.

1.3 Research questions

In our project plan (Appendix D) presented in the first phase of the project, and we defined some research questions we wanted to answer throughout the process:

- What other similar solutions exists?
- What will the different user roles in our solution be, and what are the needs for each of the roles?
- How can we make the solution as profitable as possible for both entrepreneurs and transportation companies?
- What existing solution(s)/software do transportation companies use to organize and plan routes, and is it possible to integrate this/these in our product?
- How can we make the application simple and easy to use no matter the amount of vehicles and drivers being managed by each worker in a company?
- What safety regulations are drivers legally required to follow, and how can we take these into account in our solution? Are there any additional steps we can take into our solution to make it as safe as possible?

1.4 User groups

To identify the user groups we must first set the limitations of the group. The user groups this product is intended for share interests, goals and concerns. They also operate in Norway. For this product there are two main user groups: transporters and employers.

1.4.1 Transporters

Transporters will use the application to find more transportation assignments. As a result of this, they will be able to transport more cargo, which in turn will increase their income. Today the likelihood of transporters driving with no cargo or little cargo is quite high, possibly because there is no open platform with a lot of transportation assignments available.

1.4.2 Entrepreneurs and companies

Entrepreneurs and companies will use this application because it will allow them to get their cargo transported for a lower price than it normally would have been. If the cargo does not have to be transported at a specific date, Backload would be a good solution in order to save some expenses.

1.5 Limitations

1.5.1 Time

We are all full time students, that means that we are supposed to work 37,5 hours every week. We have a total of 30 credits every semester, and this bachelor is 20 of those 30

credits. The bachelor should then take two thirds of the 37,5 hours every week. This means that every one of us should work 25 hours every week towards this bachelor.

We started working on the project the second week in January and are going to deliver the project may 16. This is about 18,5 weeks but one of these is the Easter week, so if we multiply the 25 hours every week with 17,5 weeks we get about 437 hours. We are four members in the group, so totally we will put 1750 hours into the project. Time is of course a big constraint, and we had to plan how to spend those hours wisely.

1.5.2 Cost

Costs were not a limitation. This was not because we had an unlimited budget, but rather because we did not expect any substantial costs.

1.5.3 Professional background

All members of the group are web development students and had little to no experience working as web developers. The group expected to be prepared for the challenges ahead.

1.6 Report structure

The main chapters in the report are structured in a chronological order, divided into the main three phases of our project, idea development, design and prototyping and implementation. Background, methodology and process, which do not follow this chronological order, but provide supplementary information.

1.7 Terminology

API Application Programming Interface

Aerial distance the distance of a straight line between two points

Broker A third party that arranges the contact between employer and transporter

Backloads Picking up extra load when driving "back" after a finished transportation assignment

Employer The company who needs something transported

Route The whole route a driver has to take to complete one or many transportation assignment both from our application or other sources

Sub-route The part between two waypoints of a route

Transportation assignment What an employer uploads to the application and what the transporter bids on

Transporter The company who transports something for an

UI User Interface

UX User Experience

Waypoint A route can consist of many locations along the the way. Each of these points are what we call waypoints

2 Background

In this chapter we will first introduce our product owners, then go into detail about the original idea they presented.

2.1 Product owners

Johannes Nordsveen is currently working at Ptt Communication, a company that offers a smarter and more cost effective way of communicating, replacing radios with phones or units using the same network our phone uses to communicate. The solution is called "Push To Talk" and brings functions already existing on old radios to the modern age. He have earlier worked 12-15 years with receiving and depositing contaminated gravel and masses, as well as some consulting for different entrepreneurs with contaminated ground as his field of expertise. Throughout his work with contaminated soil, he have experienced much unnecessary and ineffective use of heavy transportation. This is where he got his idea from. Rolf Martin Schonhowd plays a consulting role for Johannes.

2.2 Original Idea from Product owners

The basic idea was to make heavy transportation more effective by driving less without cargo. This will then also help the climate and leave a smaller footprint in the traffic.

Practical example

A contractor is carrying out the excavation of a large plot for residential or commercial buildings in Oslo. The contractor needs the soil that is excavated moved away, so he have made a deal with a transportation company. The trucks are fully loaded at the plot and drive out of Oslo and unload tens of kilometers outside of Oslo. On the way back to the plot they drive empty. Often there is other sites nearby that require materials like rock or gravel. The second building site have a deal with another transportation company than the first site. They have the opposite problem, they load the trucks tens of kilometers outside of Oslo and drive empty out of Oslo after unloading in the city.

Johannes Nordsveen have for some time had an idea about an application where the entrepreneurs can submit the transportation assignments, and the transportation companies can then see them and pick them up. This will cause more backloads. It will also be possible to pick up goods if you have the space, to in that way be more efficient.

The reason the transport companies and the entrepreneurs both would want to use this application is costs and efficiency. By getting more cargo the transport companies will earn more money for each truck, and they can then accept lower earnings on the

assignments. The entrepreneurs on the other hand will have to pay less for the same amount of work. An example of this can be that a truck have delivered gravel to a site in Oslo. An entrepreneur in Oslo have soil he wants transported out of Oslo. The transport company will then accept lower payment than normally to pick up the soil and transport it out of Oslo. The driver is already headed in the same direction anyway, and some pay for a backload is better than no pay. This creates a win-win situation where both earn money, the entrepreneur pays less than normal and the transport company earns money on the way back.

3 Methodology

This chapter describes the theoretical background of the methods we have used throughout the project.

3.1 Brainstorming

Brainstorming is a method used by an individual or group to come up with ideas and solutions to problems. The point of a brainstorming session is to get all the ideas on the table before discussing each individual idea. Some rules should always be followed, like no criticism or praise. Criticism and praise should happen after all ideas are on the table, not during the process (Wilson, 2013, p. 2-3). This process should not take up a lot of time, it will of course depend on the type of project, but two days is enough for a big complex project. If the ideas stop coming it is a good indication to stop the brainstorming for the project and move on (Cooper et al., 2014, p. 111-112).

3.2 Interview

Interviews are a qualitative research method used to gain insight into a specific user group. There are generally three different types of interviews: structured interviews, semi-structured interviews and unstructured interviews. The main differences between these types are as the names suggests, the structure of the interview. A structured interview has a set list of questions where the questions themselves are specific and narrow. Structured interviews are usually conducted in a controlled environment. Unstructured interviews on the other hand has a general topic where the interviewer asks open and broad questions, and is more like a conversation than an interview. An unstructured interview is often conducted in a more uncontrolled environment. Semi-structured interviews are somewhere in between the other two with perhaps a list of topics the interviewer wants answers to (Frances et al., 2009, p. 309).

There are pros and cons with all of the methods, but there are factors that influence which one you should choose. Some factors are the type of information you are after, how much time you have and how skilled your interviewer is. If you are after specific bits of information, and have had time to figure out what these are, and how to ask to get these answers, you should probably choose a more structured approach. However, if you are after more general knowledge on a subject, or do not yet know which specifics you are after, a more unstructured approach is likely better. An unstructured interview also has less preparation time but is also more difficult to conduct to its maximum potential

You are of course not limited to choosing one of these, and can combine aspects of

each. In reality most interviews will not fit fully into one category, but maybe lean a little towards one of the other categories.

3.3 Survey

Research using surveys is a method to gather quantitative data from a user group. Surveys can consist of open ended questions with a free text response, or have more quantifiable questions with for example five possible responses ranging from disagree to agree. A survey is not locked to either option and can combine the two as necessary to archive the goals of the research. The primary reason to use survey as a method is to gather the general characteristics of a large group of people (Pontó, 2015). This does not work well when only a limited amount of people takes part in the survey. Using this method the goal is often to get averages. To do this there must be sample of the user group questioned. It is still possible to get an average if there only is 3 responses, but that can hardly be called representative for the actual group of people that are researched.

3.4 Qualitative data analysis

Qualitative data analysis is about developing an understanding of a set of qualitative data. This is mainly done by categorizing, a coding process. To be able to do this process, you need to be able to find the parts of the data that is important. The process can be divided into two stages: data preparation and familiarization. The data preparation stage is about making the data as easy to consume as possible, by transcribing and formatting it properly. The familiarization stage requires you to get to know the data, consuming it. This can be done by reading transcriptions, listening to the interview again or looking through notes. When you go through the data, you categorize and code the data, and try to understand the underlying themes and meanings (Swanson and Elwood, 2005, p. 236-237).

3.5 Personas and scenarios

In the book *About Face - The Essentials of Interaction Design*, Cooper et al. (2014) describes models as a method to focus on the important aspects of something, and remove focus from the less important aspects. Personas is a model used in interaction design, which is used to create detailed models of end users. Personas are created based on research into your user group (p. 61-62).

Later Cooper et al. (2014) describes personas-based scenarios as short stories of a personas using a product. These scenarios are made to help highlight what the user needs from the product you are designing (p. 105-106).

The quality of the personas and scenarios made are affected by the quality of your research. If you interview one person and just make that one person into a persona with scenarios, it is most likely not a good representation of your user group, if you interview

five people, and do a survey for 100 people you will have a better understanding of the average user's goals.

3.6 Prototyping

Prototyping provides a possibility of filtering out the bad design choices from an early stage. If mistakes are found in the design after development it can take a lot of time to improve. If the mistakes are found when you only have sketched a paper prototype, the new iteration will not cost much time at all. Prototyping can generally be split into two categories, low fidelity (lo-fi) and high fidelity (hi-fi). Lo-fi prototypes are less accurate prototypes that take less time to make. A typical example is a paper prototype, where you do not get the benefits of the visual design aspects, but you can test how the fundamental functionality works. Hi-fi prototypes are detailed prototypes with a more thorough visual design compared to a lo-fi prototype. It is also a better representation of the actual end product. These two categories of prototyping are not set in stone but are fluid. A prototype can have aspects of both and exists somewhere in-between (Rettig, 1994, p. 21-22).

3.6.1 User testing

When conducting a user test, there are some things to keep in mind. You need users to test your application. One person conducting the test, telling the users what they should do. One or more people being observers and just taking in the feedback, taking notes and looking at how the user interacts with the prototype. If it is a paper prototype, you might need a person "simulating" the computer by moving the correct pieces of the prototype. This could be done by one of the other persons as well. The conductor should be the only one communicating with the user because it is easier for the user if there is only 1 on 1 communication. 3 on 1 communication can make the test more overwhelming (MOBGEN, 2017).

From our own empirical knowledge, we have found that overwhelming the user is in general a problem when conducting a user test. We think that to counteract this, you can have the test in a less formal setting. This will also affect other aspects of the test like causing more distractions, but we believe finding the balance is important.

3.7 Agile development

Agile as a term got its foothold in 2001 when 17 software engineers got together and created the "agile manifesto". This manifesto provided a set of four main values in the agile methodology (Schmidt, 2016, p. 15):

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan (Schmidt, 2016, p. 15)

The main goal of agile development is to spend more time on the actual improvement on the end product, and use less time on extensive planning and preparation for a perfect process (Schmidt, 2016, p. 15). Improving communication and reducing the time between a decision and seeing its impact are important aspects of agile development as well. To improve the communication there are multiple things that can be done, literally moving people closer to each other is one way. Working to improve the teams comradery to make communication flow easier. Another way to improve this is making the communication verbal instead of written. To reduce the time from a decision to its impact you can work incrementally, or have the people who will notice the potential problem close at hand, preferably in the team (Cockburn and Highsmith, 2001).

3.7.1 Scrum

Many people think of Scrum as a development method, but it is actually a project management framework. It is often used by smaller teams and is based around people having specific roles and works in small periods of time called *sprints*. The project as a whole has one big backlog, which consists of all the tasks needed to complete the project. For each sprint, tasks from the backlog is moved to the sprint backlog, and the team attempts to complete all the tasks within the sprint window. Every sprint starts and ends with meetings where the changes are presented to the product owner, and the new sprints are planned and improved based on experiences from the previous sprints (Schmidt, 2016, p. 16-17).

4 Idea development

In this chapter we will go through the iterations of the project idea.

In the first third of this project we had a lot of changes in our project, like what the scope should be and which part of the transportation industry should have the main focus. During this phase the base idea went through two iterations that fundamentally changed the idea.

In our initial meetings in the start of the project we did a lot of planning and preparation for the project. We also started doing research early in the process. This research can be split into two main categories, research of existing solutions and research of the Norwegian transport industry. We wanted to find the most similar competitor here in Norway, and the biggest global competitors. There were also some things we needed to learn about the industry, and we wanted some statistics to back up the problem presented in the original idea. One group member conducted an unstructured interview with an acquaintance that works as a truck driver to get some insight about the industry.

4.1 The Norwegian transportation industry

In the third quarter of 2017 there were 67,8 million tons of goods transported by Norwegian trucks. The average length a ton is transported went up from 66 to 72 kilometers compared to the third quarter of 2016. In the third quarter of 2017 trucks traveled a total of 469 kilometers in total. 137 kilometers was driven without any load at all, that is 29,3% of the total kilometers driven in the third quarter of 2017. One year before, in the third quarter of 2016, it was 28,8% of the total kilometers that quarter. That means that in a year it increased by 0,5%. When Norwegian trucks traveled outside our borders it was only at 21% ([Økt transportarbeid](#)).

There are strict regulations about how much a driver can drive without taking breaks. These regulations cover everything from how much you can drive continually without any breaks, to how much you are allowed to drive during a whole week. There are some options where you can for example split your 11 hour mandatory rest during a 24 hour period into a nine hour rest with one three hour rest at some other point in the 24 hours. If you need to abide by this law you need to use a device in your truck that will track when you are driving and resting ([Statens vegvesen, 2018](#)).

4.2 Competitor research

After we had some some initial research into possible competitors and similar solutions we put our findings into a document(Appendix G) so we could present them for product owner. This document consists of two parts. In the first part we present the existing solutions we want to highlight with screenshots and short descriptions of what the application does. We present six similar solutions. Five of these are basically the same basic idea as the original idea we had. A marketplace for transporters to get transportation assignments. None of these five are in Norway though so they are not exactly competitors, at least not yet. The last one is called Nimber and is Norwegian, but its not exactly the same. Nimber is a marketplace for anyone that needs something transported, and the transportation can be done by private people and their car, not only professional businesses. In the second part of the document we wrote down the problems we had found about the original idea and possibilities and our suggestion for the way forward.

4.3 First iteration

The second part of the document was more of a letter to our product owner. We presented three main problems we had found with the original idea (Appendix G).

Problem 1: Drivers plan their routes, so there is little room to pickup an extra transportation assignment. Most of the time you either have to take a break by law, or you have another transportation assignment booked too soon after so the time it takes to do pick-up and drop-off is too big. Even if the extra transportation assignment is exactly in your path, which in itself is unlikely.

Problem 2: The truck driver himself will not be the one earning more money, unless he has a sole proprietorship. He is driving the company's truck with the company's fuel, and can not use this to drive off route for his own transportation assignment.

Problem 3: Due to safety reasons the driver should not operate an application during driving. Even if the guidelines say that the application should only be operated while standing still, the reality is that drivers push these limits, and would even more if there had been room for earning extra money. This is not an opportunity we want to provide, because of the safety reasons.

After presenting the problems we had found, we also provided two solutions, or ways we could move forward (Appendix G).

Solution 1: An application for planning and organizing routes and trucks, where you also can get more transportation assignments

Solution 2: A niche application that is mostly like the original idea, but will only work for sole proprietorships and very small companies(1-3 trucks)

We said which solution we would like to pursue and asked the product owner which one he would like us to pursue. The product owner agreed with us and went on with the idea of a planning application for the office workers in a transportation company. After landing on an iteration we finished our project plan with the new scope.

4.4 New problems

After we had landed on the new iteration of our idea we started by getting some of our own ideas down on paper. We first sketched individually to get each persons thoughts out(Appendix M), then went on to discuss the ideas we each had. We again started to see problems with the new iteration:

Problem 1 the scope is too wide, so we will not have time to put as much time as we want to into each aspect.

Problem 2 the transporters do most likely not want to use multiple platforms for planning, and therefore have to move over to our application to use it, which we think would result in less customers.

These problems came pretty fast after the first iteration was complete, so we agreed we needed to learn more about the industry and then make changes based on that. While sketching we also wrote down questions we came across for our target audience. After getting some questions down on paper we started discussing the best method to get answers to these questions. Interviews were the first thing that came to our mind. Since we did not have a lot of experience in the field we would want answers with more possibility of follow-up questions and more explanation if they made assumptions about how much we knew. We also wondered about some more quantifiable things like how many drivers/trucks there are per office worker that handle planning and organization or how they organize the routes, how big the firm is, total amount of trucks and trailers. Therefore we agreed on conducting both interviews and sending out surveys.

4.5 Survey

We ended up with around seven questions for our survey(Appendix C). We then created a Google form with the questions and emailed it out to 40-50 transportation companies around Norway, our product owner also sent it to his contacts. Even though 50+ companies might be big enough of a sample to get a good idea of the reality, it does not mean that 50+ will answer, but we hoped at least 10 would. In retrospect this was too ambitious of us. We got a total of three answers, which we could get some limited information out of. It was however far from enough to get a good indication of the general state of the transportation industry standards so we could not use this data as a reliable source.

4.6 Interview

For our interviews we contacted the transportation companies we found around Gjøvik. We sent emails to five companies telling them about our project and asking if they were willing to give us a interview and got interviews with two of them. At this time we had 15 questions for our interviews(Appendix C). These included all the survey questions and also had questions going into more specific aspects of the everyday work in the companies.

We made standard consent forms for our interviews(Appendix E) where the user agreed or disagreed to us recording the interview for transcribing it afterwards and to us publishing said transcribing in its whole in the project report. We transcribed the first interview we did, but this took quite a lot of time so we chose to not transcribe the other interview, and just discuss and analyze the results we had found. After the interviews we realized the wording on the agreement to publish the transcription in the report was unclear. We agreed that if we were to publish it we would need to confirm with the transportation companies it concerned.

Initially we started a qualitative data analysis method described in 3.4. Our main goals with the interviews was to learn more about the industry and how their day to day activities are run. After some discussions we came to the conclusion that we would not get enough out of the method to justify the time spent. We used the details we had learned during the interviews to create some personas from different types of companies (Appendix B). The main differences in the companies is the size of them, which we learned from the interviews played a big part in the usefulness of this type of application. Based on these personas perspective and our own thoughts and experiences we discussed the scope and idea for our project.

4.7 Second iteration

These discussions resulted in the second iteration of the project idea. In this iteration we have completely removed aspects of the idea. The planning part where the office workers has control over their fleet is removed. What we ended up with then was only the marketplace part of the application where transportation companies can get extra assignments, but everything after that follows their own protocols and routines. We only want to serve as a broker for the transporter and employer. We also removed the driver view, because if we do not have a planning view, then they will use their current methods of planning, and will use their current method of sending routes to their drivers.

Another change was a reduction of the intended target audience. Initially the idea was that any transporter can use this application, and it will still be possible. However, we learned from our interviews that bigger companies will have less use of an application like this. One argument for this that we found when we started discussing it was that a smaller company has less trucks and most likely less available assignments, and because

of this fewer possibilities to maximize each drivers route.

From the start of the project the regulations around breaks while driving was something we found important and wanted to include in our solution. In this iteration however it would have resulted in a lot of input for each route the user would search. There is not enough to gain from it to justify asking the user to input this information so we ended up scrapping it.

5 Design and prototyping

This chapter covers all the functionality we discussed up until this point in the project, and throughout the prototyping process. It also presents the finished digital prototype and its testing process. Before we started the prototyping process we redefined our project and went into more detail about the scope, and other factors that will have an impact on the end product(Appendix F. The scope defined in this document evolved during the prototyping process and will not completely reflect the finished digital prototype.

5.1 User experience

The terms User Interface and User Experience are commonly mixed up and incorrectly used interchangeably, over time this has made it hard to define the actual meaning of UI and UX, and where we draw the line between them. Depending on what you read or who you talk to, you might get quite conflicting definitions of UI and UX. Don Norman, who is credited for coining the term User Experience, said it himself in an interview from 2007 that the word User Experience no longer had any special meaning because of how often it is misused ([Merholz and Norman, 2007](#)).

When the term User Experience is brought up in our discussions of design, we are referring to Don Norman's definition: "User experience encompasses all aspects of the end-user's interaction with the company, its services, and its products" ([Norman and Nielsen](#)). Which implies that User Experience is not simply the emotional reactions of a person when interacting with the application itself, it extends beyond the present moment and includes all responses that take place in the users journey.

5.2 Features and functionality

One of the main focus areas for us have been to come up with a product that is both easy to use and understand. In order to facilitate for a good user experience, we have had numerous discussions regarding features and functionality that could potentially be added into the system.

5.2.1 Search for transportation assignments

In our opinion, one of the most crucial aspects of Backload will be the solution used for finding transportation assignments. In most cases we imagine that a transporter will only be interested in looking through transportation assignments that meet certain criteria. The vast majority of transportation assignments will therefore be irrelevant. As an example, a transporter that only drive in and around Oslo will not be interested in taking

any transportation assignments elsewhere. The main idea of Backload is to decrease the amount of time trucks are driving around with empty cargo or with room for more. We therefore assumed that a transporter will have a certain route and schedule in mind, for one or more of their trucks. This is why, in order to show only relevant transportation assignments, we wanted the user to conduct a search, and then show results based on that. If the user does not conduct a search all transportation assignments will be shown.

Plot a route

Perhaps the most important factor to take into consideration for a search is the geographical area that should be included in the results. As mentioned earlier, we assumed that a typical Backload user will have a certain route in mind when looking for transportation assignments. We therefore wanted to limit the search results to only show transportation assignments that are somewhat nearby the original route. We were thinking that this would be based on how close the pickup and delivery points of the transportation assignments are to the user's original route. We did not know the best way to measure this. Simply measuring the linear distance in kilometers would not necessarily give logical results since there could be obstacles like a mountain or an ocean in the way, which could demand a lot of time to circumvent.

The topic of how to measure closeness is one we discussed a fair amount. One solution that we liked is to calculate how much time, in hours and minutes, it would demand to deviate from the original route and drive to the pickup point of a transportation assignment. Being within a certain time limit (for example 60 minutes) would then be used as a criteria to be shown as a result. We also felt like the user should be able to change the set time limit since this would give the user more control. However, this proved to be too technically challenging and we ended up implementing another solution due to time constraints.

We were thinking that in order to conduct a search, the user should plot a route. The search results will then only show transportation assignments that are nearby the plotted route. In order to create a good system for plotting a route, we researched how other mapping services that do this well have solved this task. Some of the services we have looked into are Waze, Bing Maps and Google Maps.

According to Waze, their service is the the world's largest community-based traffic and navigation app ([Waze](#)). After looking into Waze, we got the impression that it is mainly a mobile app meant to be used while driving. This is not really what we were looking for since we moved away from including a driver view in the system, and are solely focusing on finding and submitting transportation assignments. However, Waze also have a website with a "live map" where one can enter a start point and destination point. This is similar to what we wanted to do with Backload. When submitting a search in Waze, one or more color-coded routes are shown as results on the map. Information about a route can be found in a list next to the map, with an icon of the same color of

the route. Figure 2 shows how this looks like for a search from Gjøvik to Molde.

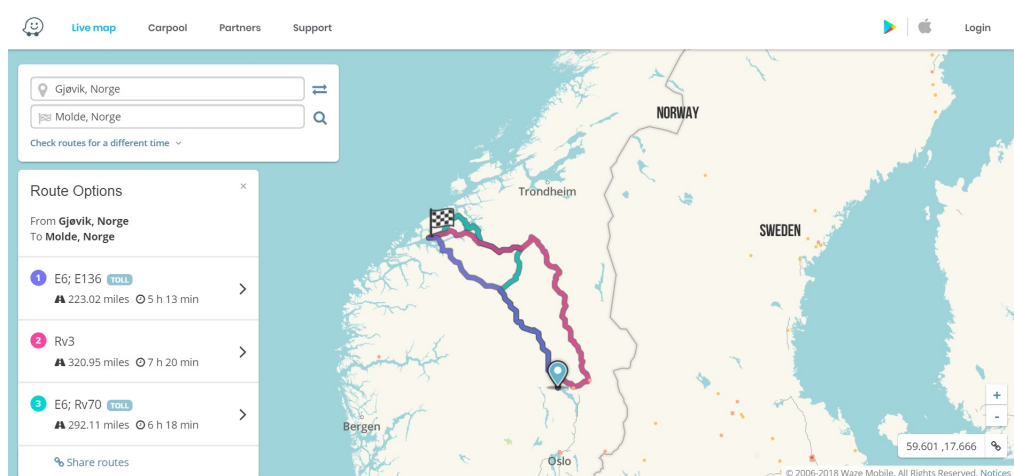


Figure 2: Example of a Waze live map search for the route Gjøvik - Molde

Both [Bing Maps](#) and [Google Maps](#) are map services that work well for planning a route. Just like Waze, both of these services show useful information like estimated travel time, if the route includes tolls, if there is light or heavy traffic etc. One big difference between Waze and these two map services is that Waze can only plot a route between a start and destination point, whereas Bing Maps and Google Maps offer the functionality of adding waypoints along the route.

When studying the layout and functionality of Bing Maps and Google Maps, we found that they are in fact quite similar. Figure 3 and Figure 4 show a search made for the route Gjøvik - Molde - Trondheim in Bing Maps and Google Maps, respectively. Both services have a sidebar to the left of the screen, overlaying the map which takes up the whole screen. This is also very similar to how Waze have solved their layout. For Bing Maps and Google Maps, the placement of the UI elements and the functionality present in the sidebar, where one can plot a route, is almost identical.

They both provide the option of choosing method of travel at the top (travel by car is the only one relevant to us) by clicking on icons. Just like Waze, the route is plotted by writing into input fields for start and destination points, and they are shown on top of each other, in the order of start to destination. All three services also update the position and zoom level of the map when submitting a value for one of these input fields. All three services also provide the option of reversing the order of start and destination point.

In addition to this, Bing Maps and Google Maps have a plus icon underneath the destination point that can be clicked to add another point. A new input field will then appear, which will function as the new destination while the previous destination will act as a waypoint. Bing Maps and Google Maps also provide the option to remove a point (if three or more) by clicking a delete icon to the right of each input field. Both services also offer the functionality of rearranging the order of the waypoints (including start and

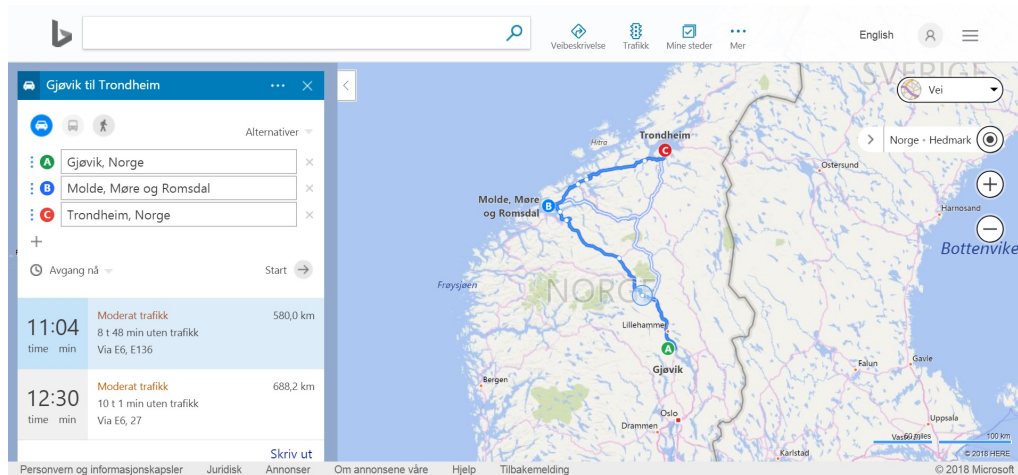


Figure 3: Example of a Bing Maps search for the route Gjøvik - Molde - Trondheim

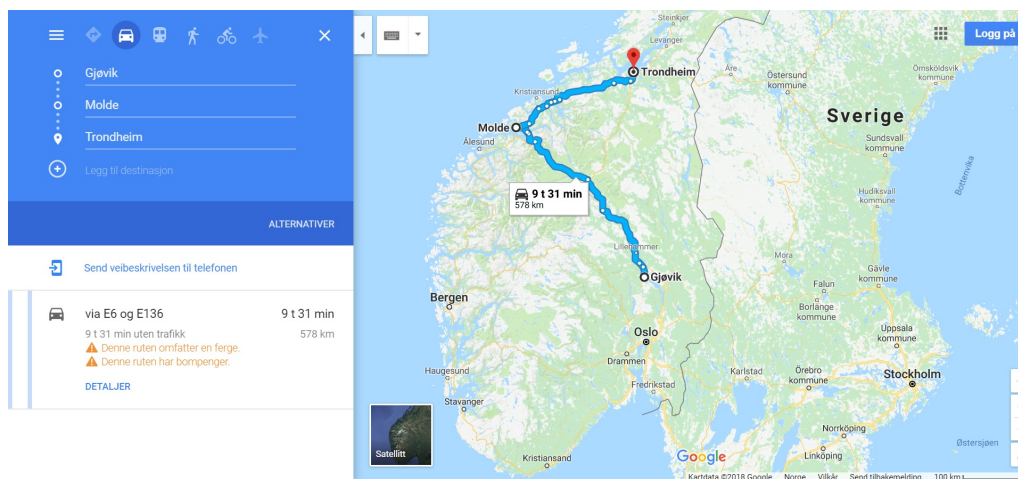


Figure 4: Example of a Google Maps search for the route Gjøvik - Molde - Trondheim

destination) of a route by using drag and drop.

After we had studied these map services, we discussed what kind of functionality we wanted to implement into Backload. We have based much of the functionality and layout off the sidebars that Bing and Google use. One thing that we agreed on early was that we wanted the user to be able to plot a route with multiple waypoints. The interviews we conducted of the target audience told us that it is normal to either drop off some cargo or pick up additional cargo at multiple points along a route. With this in mind, we assumed that a typical truck often will drive a route via multiple locations before reaching the destination. This is the main reason why we thought this would be an important feature. We liked the way Bing and Google positioned the input fields on top of each other, in order from start to destination, with the option of adding additional points. From our own experience using Google Maps over many years, we felt like this was an easy and intuitive way of plotting a route from start to finish. We therefore decided to go

for the same approach. Other features from Bing and Google that we liked and wanted to implement into Backload was the options for removing a point, and reordering the points by using drag and drop. These are actions that make it easier for the user to make changes to a route.

Filtration

In addition to geographical distance, there are other key factors to take into consideration for achieving the best search results possible. We were thinking that in order to narrow down the results, it would be beneficial for the user to be able to filter out results based on these key factors.

We imagined that one of these factors would be the period of time that the user will be interested in taking on a transportation assignment. We were thinking that ideally, when conducting a search, the user should select both a start and an end date. The search results will then only include transportation assignments that fall within the selected time period. In other words, only results that have a pickup date that is set later than the selected start date will be shown. Correspondingly, only results that have a delivery date that is set earlier than the selected end date will be shown. The idea here was that the user should be able to change the start and/or end date at any time in order to control what search results are being shown.

Another key factor we have considered for conducting a search is the type of cargo. From what we have learned by researching the transportation industry and from the interviews, we knew that certain types of trucks are used to transport certain types of cargo. In other words, a truck that is capable of transporting one type of cargo cannot necessarily transport other types of cargo. For example, a truck transporting a load of gravel will most likely not be able to pick up an additional cargo of pallets.

We assume that a user often will have a specific truck in mind when looking for transportation assignments. We thought that flooding the search results with transportation assignments regarding cargo that the user is unable to do anything with would cause frustration. We felt like if the user could instead limit the search to only contain results regarding the types of cargo of interest, that would make for a much better user experience. This is why we decided that a filtration for type of cargo would be necessary. We came up with three types of categories for type of cargo: (1) pallets, (2) masses and (3) other. More categories could of course be added, but we thought that the majority of transportation assignments would be either pallets or masses. We therefore felt like these two categories, in addition to another category for everything else would be sufficient. We briefly discussed if the user should have the option to select more than one category, or if they should be exclusive. In the end we did not see any major benefits of preventing multiple categories from being selected.

In addition to these key factors, we also wanted to make it possible for a user to hide parts of the plotted route. We refer to a sub-route as a part between two waypoints of

the plotted route. The main reason for allowing the user to hide sub-routes is that even though the user has plotted a route with multiple waypoints, the user might know for certain that a truck cannot take on any additional cargo during some parts of the route. We wanted to give the user the option to prevent transportation assignments, that are nearby these parts of the route, from showing up in the search results. The idea here was to have a button for each sub-route, that could be toggled on or off, to decide if transportation assignments nearby a specific sub-route should be shown in the results or not. We felt like giving the user more control by allowing the removal of irrelevant search results would make for a better user experience.

5.2.2 Display search results

We had many discussions about how we should display the transportation assignments when conducting a search. Since the core activity of conducting a search is plotting a route, it was clear for us that some sort of map to show that route was necessary. In addition to this, we also needed to decide on how we should display the search results. From early on we had envisioned some sort of split between the map view and the display of the search results.

Map

When trying to figure out how the user should plot a route when conducting a search, we looked into other map services. More specifically Waze, Bing Maps and Google Maps. See [5.2.1](#) for more details. One thing that we felt like all three map services did well was to give feedback to the user when submitting a value for an input field. They continually gave feedback to the user by re-positioning the map and changing the zoom level to an appropriate value where the whole route was visible. We felt like this makes for a good user experience since it allows for a quick and effortless confirmation that the correct location was entered. Providing this constant feedback while plotting the route was definitely something we also wanted to do with Backload.

We wanted the map to offer more than simply showing the plotted route for a search. We discussed what kind of functionality the map could provide, that would be beneficial for the user. This was when we came up with the idea of displaying the routes for the search results on the map, in addition to the plotted route. The idea here was that the plotted route would be highlighted, while the routes from the search results would be greyed out and less visible. Our thoughts behind this idea was that this could make it easier for the user to visualize where the route of a potential transportation assignment would be in relation to the plotted route. We quickly realized that we should not show all of the search results on the map. The reason for this is that we felt the map could easily become cluttered if many routes were displayed at the same time, especially if the routes were close to each other. We therefore came up with the solution of only showing a certain amount of routes at the same time, for example the three first search

results. However, we felt like simply showing these routes on the map would not achieve much if the user could not easily find more information about them. This is why we wanted to make them clickable. The idea here was that when clicked, the user would be taken directly to the search result with all of the information regarding that route.

Table

The immediate idea we came up with for displaying search results was a table where one row would contain data for one transportation assignment. The impression we got from the interviews with the target audience also lead us to believe that a tabular display of data would be well received. This is because the transportation companies that we talked to were very familiar with working with tabular data. We therefore thought that a table would be an intuitive and recognizable way for them to search through large amounts of data quickly. We did however encounter some challenges when trying to display the information that we needed. There was just way too much information for each transportation assignment that everything could fit nicely into a row. The smaller the screen, the bigger this problem would become. This is why we also looked into alternative ways of displaying the data.

A trend that has become quite popular lately is the use of so-called cards, and this is the approach we considered going for instead of a table. A card can be described as a rectangle including images, texts, links buttons etc, often used as a teaser to detailed information ([Best practices for designing cards – UX Planet, 2017](#)). In an article written for Designmodo, [Cao \(2017\)](#) explains that cards are small containers of information, with each card representing its own singular thought. Even though a card can hold all types of content, they should all fall under a single unified theme. One of the big advantages of using cards is the fact that they work very well in a responsive environment since they are often displayed in a grid. One can easily rearrange the cards based on how many cards that can fit in the width of the screen.

The idea we envisioned was a grid with cards, where the amount of columns would change based on the screen width. The crucial information like pickup time and location, delivery time and location and the cargo would be displayed in the card itself. The card would be clickable and open up either a modal or expand the card in height and/or width to show the rest of the information. We quickly saw multiple problems with using a card based grid over a table. First of all, we thought it would be more difficult for the user to compare transportation assignments to each other when they are displayed with cards. In a table, the data for a transportation assignment can be clearly separated by columns like "pickup date", "type of cargo", "bid deadline" etc. With cards, we thought this would be a lot more difficult, since there are no column headers to look at and directly compare.

Another problem we saw by using cards is that each card would take up more space in height and width than a table row. Our thoughts here was that a table would allow the user to see more transportation assignments at once, while if using cards the user would

have to do more scrolling. The main reason that we looked into using cards in the first place was because it was difficult to fit all of the data for a transportation assignment into a row. However, we figured out that we would encounter the same problem when using cards since we could not realistically show all of the data inside the card itself.

We ended up taking the idea of expanding the card to show additional info in combination with the original idea of using a table. The idea here was that when a row is clicked, it would expand in height to show additional content. This would allow us to show the most crucial information in the row itself and additional information like detailed contact information, information about previous bids, and the possibility to submit a bid. See [5.2.3](#) for a thorough explanation of how we would like bids to work.

For the table we also wanted to make it possible to sort the results based on certain categories. We quickly decided that we would take a common approach that we had seen used a countless amount of times. In this approach, column headers are made interactive by making them clickable. We wanted it to be possible to sort results in either ascending or descending order based on the clicked column header. Depending on the values stored for the column, the sort would be done either alphabetically, by amount or by date. From our own experience, we felt like this method of sorting is clear to understand and easy to use.

Show transportation assignment on the map

One feature we really liked with Waze live map was their use of color-coding. Multiple routes would be shown on the map, where each one was highlighted with a different color. This color would directly correspond with an entry in the list containing information about the routes. We felt like this way of creating a connection between route and information made it easier and faster to find the relevant information. We were inspired by this, and decided to do something similar.

We wanted the user to be able to view any of the transportation assignments from the search results on the map. We would like for each row in the table to have a button that could be clicked, resulting in the route being displayed on the map. We wanted the user to be able to select multiple routes to be shown on the map at the same time. Our reasoning for this is that we thought this could be a nice way of comparing the routes for different transportation assignments. In order to clearly distinguish multiple routes from each other we wanted to make use of color-coding. Our thoughts around this is that the button should be a toggle (on/off). When toggled on, the button should be filled with a color. At the same time, the route should be displayed with the same color on the map. If the user clicks on the button for another row, the same thing would happen but with another color.

5.2.3 Bidding wars

A key aspect of the Backload system is to connect employers with transporters. Initially, we thought about simply presenting the transportation assignments and providing the contact information for the employer. It would then be up to the transporter to make contact with the employer and make a deal internally. However, we quickly agreed that we wanted to simplify the communication between both parties.

Our thoughts around this is that transporters will only be interested in transportation assignments that match certain criteria - in addition to being in the vicinity of one of the routes being driven by one or more of the transporter's trucks. We assumed that the most important criteria here would be the price of payment. The transporter must make a decision whether or not taking a transportation assignment would be worth it, based on the amount of time and costs associated with deviating from the truck's original route. On the other side of things, the employer would most likely be interested in finding the transporter who could get the job done for the cheapest price.

We decided that implementing some kind of bidding functionality would be a possible solution for pleasing both parties. We have spent a lot of time discussing how a "bidding war" could work, and have explored numerous ideas. In order to land on a solution, we needed to come to an agreement on a number of things.

We needed to specify what a bid actually is and the information that would be needed from both parties in order to make a bid. Ultimately we decided that the employer would need to set a deadline for how long bids can be submitted for the transportation assignment. Within this deadline, a transporter can make a bid of any amount.

Whenever a bid is made, both a suggested pickup time and delivery time must be provided by the transporter. A suggested pickup and time must include both the date and the estimated time of arrival, in hourly intervals. We also decided that it should be possible for the transporter to add a short text containing extra information. Our reasoning behind this is that delivery of cargo is not always straightforward and some delay is to be expected at times. Estimating the time of pickup and delivery to the closest hour is not always feasible. An example could be that a transporter selects the estimated pickup time of 14:00 on a specific day, but is actually available during the window of 14:00-17:00. This could be easily explained in a short text.

A transporter is allowed to submit multiple bids. This means that various transporters can make bids of the same amount for the same assignment. This is an intentional decision made by us to not force transporters to bid lower than the others. We made this choice for two main reasons: The first one being that we did not want the lowest bid to automatically win the bidding war when the deadline ended. We felt like ultimately, the choice of transporter should be made by the employer based on price of the bid and the reputation of the transporters. See [5.2.4](#) for details regarding reputation. The second reason is that, as just mentioned, we wanted transporters to be able to submit

multiple bids. This is because a transporter would usually own more than one truck. It is not unreasonable to assume that a transporter would have multiple trucks driving in the same area as the transportation assignment, but at different times. We therefore felt like it would be beneficial for the transporter to be able to submit bids for multiple pickup times (and prices) without having to outbid themselves. One of the big talking points we encountered when discussing bids was whether or not a transporter should be able to withdraw a bid. There are pros and cons for allowing bids to be withdrawn. During our discussions there were especially two arguments for allowing bids to be withdrawn.

Firstly, we assume that a transporter might be interested in multiple transportation assignments, which are well suited for a specific route for one of their trucks. If one is automatically committed to a transportation assignment when making a bid, it would then be a bad idea for the transporter to make bids for two transportation assignments that would fit well for that truck. The reason for this is that if the transporter wins both transportation assignments, only one of them can be completed since they overlap. Secondly, we assume that a transporter might make mistakes when submitting a bid. For example choosing the wrong pickup/delivery times or the wrong price. It would not make for a good user experience if the user is then forced into completing the transportation assignment based on conditions that were wrongfully submitted.

The main argument we came up with for not allowing transporters to withdraw their bids is security for the employer. We felt like it would be crucial for an employer to have confirmation that the transportation assignment will actually be completed. If transporters are allowed to withdraw their bids, the employer could potentially end up with zero bids for an assignment that originally had multiple bids. We thought this would be very confusing and frustrating for the employer, which might have relied upon one of the bids that were withdrawn to complete the assignment at a specified time.

Taking both sides into the consideration, we decided that once a bid has been submitted, the transporter is committed. Specifically, this means that the employer can expect the transporter to complete the transportation assignment for the price and time for the bid that was submitted. However, in case of any errors being made when submitting the bid, the transporters will be allowed to edit or withdraw a bid for a certain amount of time after the bid was made - for example an hour. After this time period, the transporter will no longer be able to do any changes or remove the bid. The contact information for the employer (and transporter) will always be available, so communication between employer and transporter will always be possible if needed.

We also needed to agree upon how a bidding war should end. We briefly discussed whether or not the employer should be able to end the bidding war early by accepting one of the bids. We came to the decision that the employer should not be able to do this. Our reasoning here is that the employer should already have set a deadline which works well. We did not see a big benefit in ending the bidding war early since the deadline should

already fit, and better bids might be submitted later. Sticking with a final deadline would also allow the transporters more freedom in when to submit their bids, without fear of the transportation assignment suddenly no longer being available.

After the deadline, the employer must make a decision. We agreed that the employer should be allowed the option to decline all bids, and therefore not giving the transportation assignment to any of the transporters. The reasoning behind this is that the employer might not always be satisfied with the conditions set in any of the submitted bids. However, we felt like it was crucial that the transporters who had been participating in the bidding war should receive feedback about this choice. To avoid a situation where the transporters have to wait in uncertainty for a long time, we decided that the employer would have a set amount of time (for example a window of 6 hours) after the deadline to either accept a bid or decline all bids. One potential backside we could imagine here is that the employer might not be available to make the decision in this time window. A potential solution for this would be to give the employer the choice of automatically accepting the lowest bid after these 6 hours have passed. This is not something we thought would be a big issue, since the employer is in full control when setting the deadline of the transportation assignment, and therefore also in control over when the time period for accepting/declining bids will occur.

5.2.4 Reputation

The idea of displaying some kind of reputation for both transporters and employers struck us early on in the idea development process. Since we want multiple transporters to be able to bid the same amount, one could easily end up in a situation where two or more transporters end up bidding the exact same amount. If the bids have different pickup/delivery times, the employer might prefer the conditions in one bid over the other(s), and make a decision based on that. However, we feel like this is not a very beneficial way for the employer to choose between transporters. In some cases the pickup/delivery times might also be the same and then the employer would pretty much have to choose on random.

We felt like including some indication of trustworthiness and professionalism for both the transporters and employers would be a good way to distinguish them from each other. The greatest benefit we can see is for the employer to now have a much better foundation to make a decision between two or more bids. An example to illustrate this could be that a transporter with a high reputation makes a bid for 2000 NOK, and another transporter with a bad reputation bids 1500 NOK. In this scenario, we imagine that the employer might be interested in choosing the transporter with the higher reputation even though the price is higher. This is because the reputation would indicate that the transporter is to be trusted. On the other side of things, a transporter might be interested in bidding on a transportation assignment submitted by an employer with a high reputation, but might not be as inclined to do so if the employer has a bad reputation. All in all, we

feel like showing some kind of reputation would be beneficial for both transporters and employers in their decision-making process.

We then quickly decided that a rating system would be our approach for implementing reputation. We had numerous discussions about how such a rating system should function. For a long time the decision was going to be either a star rating system or a thumbs up/down system. The reason that we ended up mainly discussing these two types of system is that from our own experience they are commonly used in popular and professional services.

We searched the web for examples and inspiration from other services. Amazon and eBay were two of the services that caught our attention. They both make use of a star rating system. Amazon allows the buyer to rate the seller on a scale from 1-5 stars, where 5 is the best. The average rating of the seller will then appear alongside their name on the site ([About Comments, Feedback, & Ratings](#)). We also looked deeper into eBay's star system, which is actually quite advanced. Ebay allows the buyer to leave a general rating of either positive, neutral or negative for a transaction. A seller gains 1 point for a positive rating, nothing for neutral and -1 for a negative one. This makes up what eBay call a feedback score. This score is shown next to the sellers name and has a different kind of star (color and shape) based on the total score. At least 10 ratings is needed for receiving a star ([Feedback scores, stars, and your reputation](#)). In addition to this, buyers can rate specific aspects of the transaction, like communication and shipping time, with 1-5 stars. The average rating for each category is then calculated and displayed for the seller. 10 ratings is needed within each category to get a rating ([Detailed seller ratings](#)).

We also searched the web for popular services that make use of a thumbs up/thumbs down rating system, and we got the impression that Netflix is currently the one making the most waves. Netflix previously used a 1-5 star rating system, but they switched over to a thumbs up/down system in 2016. This change was the topic for a lot of articles we found on the web, with a lot of mixed opinions about the decision. Cameron Johnson, Netflix director of product innovation, explains that the change was made in order to improve the suggestions for new titles to watch. Over a year of testing has shown that while Netflix has used stars to personalize suggestions, many of their members were confused about what they do. Johnson explains that when people see thumbs, they are aware that they are used to teach the system about their tastes ([Johnson, 2017](#)).

In the end we actually moved away from a rating system. From what we have learned we felt like a thumbs up/down system is best suited for services where personal recommendations/suggestions are important. This is not a part of our vision for Backload, so we therefore decided that a thumbs up/down rating system would not be a good fit. We also felt like an advanced star system like eBay is using would be way to overkill for what we wanted to do with Backload. We simply wanted a clean and simple way of showing the reputation for a user. This meant that the simple 1-5 rating system Amazon uses

was the only realistic approach remaining for us to take. However, we became in doubt whether or not users would be interested in leaving ratings in Backload at all. Backload is not likely to be the main system for finding/submitting transportation assignments, but more of an extra service to use in addition to what they are already using on a regular basis.

With this in mind, we realized that showing statistics would achieve much of what ratings do, with the benefit of not demanding users to put in any effort at all. For the transporters, we will show the amount of completed assignments next to their name in the bids. A high number would indicate that this is an experienced transporter that can be trusted. A low number would mean that the transporter is either a relatively new user or that he is making unreasonable bids, resulting in few transportation assignments won. For employers we will show the amount of transportation assignments that they have submitted. We liked the idea that eBay uses with needing to have 10 ratings to get a feedback score, and imagined that a transporter or an employer would have to complete a certain amount of transportation assignments in order to get a reputation (numbers shown) at all.

One problem we potentially saw is that the rich might get richer, meaning that transporters with a high number will be prioritized and keep growing their numbers, while the transporters with fewer numbers will have to make lower bids in order to compete with their reputation. We did not see this being as big an issue for the employers though, since their transportation assignments is what the transporters are competing for in the first place.

We were in doubt whether or not a simple statistic like this would be good enough, and have talked about maybe showing color coded icons next to the transporters and employers. We were thinking green for a good reputation, nothing for neutral/normal and red for a bad reputation. We imagined that transporters and employers can increase their reputation every time an assignment goes smoothly. We were also thinking that a transporter could lose some reputation by submitting a bid, but then when the employer makes contact after the deadline the transporter says no to the job. Similarly, an employer could lose some reputation by not accepting or declining a bid within the set time period after the deadline. This is not something we have focused further on though.

5.2.5 Favorite transportation assignments

We felt like a situation where a transporter is looking through search results but is not interested in immediately submitting a bid would be likely to occur. We therefore thought it would be beneficial for a user to easily be able to find specific transportation assignments at a later time. This is why we wanted to include the option to favorite or save a transportation assignment for later. When favorited, the transportation assignment would be added to the user's personal selection. The user could then look through this selection at

a later time, instead of going through the search process again. This could save the user valuable time. When conducting a search, each result is displayed as a row in a table, see [5.2.2](#) for details. We wanted each row to have a button that could be clicked in order to save that transportation assignment to the user's favorites. The button would be a toggle. When toggled on, the transportation assignment would be added to the user's favorites. When toggled off, it would be removed

5.2.6 Overview

For a typical user in Backload, there could be a lot going on. This is why in addition to the main page that revolves around search and display of transportation assignments, we wanted to provide an overview page. The idea here was that this page would provide all information that is relevant for a specific user. We were thinking that the overview page would make use of a sidebar with tabs that could be clicked. Only one could be selected at a time, and the information being shown would rely on which tab is selected. We came up with four tabs. The first for showing active transportation assignments, the second for ongoing bidding wars, the third for the user's favorites and the fourth for the user's history (record).

Active transportation assignments

First of all, we felt like the user should have an easy way of finding active transportation assignments. By "active", we mean a transportation assignment that the user has won, but that has not yet been completed. What we would like to do here is to display all of the user's active transportation assignments.

Ongoing bidding wars

We also felt like it would be important to provide easy access to the bidding wars that the user is involved with. The idea here was to display all of the transportation assignments that the user had submitted a bid for, and where the bid deadline is not over yet.

Favorites

In [5.2.5](#), we talked about how we wanted to give the user the option to favorite a transportation assignment. This is where all of the user's favorited transportation assignments would be displayed.

History

We were thinking that it would be beneficial for the user to be able to look through all of the transportation assignments that he has completed. This could be useful if the user would like to see the details of a completed transportation assignment, or to get in touch with the employer. This is where a record of all of the user's completed transportation assignments would be displayed.

Notifications

We imagined that for an active Backload user there would be a lot going on, especially in regards to bidding wars. A user can go to their overview page to look at detailed information about all of the transportation assignments that he is involved with. However, we felt like it should not be necessary for a user to do this in order to stay updated on the status of these transportation assignments. This is why thought some sort of system for notifications definitely should be added into Backload.

UX Planet is a blog website dedicated to the topic of user experience. They describe a notification as an act of bringing something to the notice of the user. More specifically there is a type of notification most often referred to as push notifications. UX Planet explains that all kinds of notifications can be thought of as a push notification for the simple reason that they are pushed through by the system. There are two types of push notifications. The first one encourages immediate action from the user. For example an email alert or a request to change a password. The second one simply provide information to the user. There is no need to take any immediate action on it. For example a weather update ([How To Design Notifications For Better UX, 2017](#)).

Webdesigner Depot is another blog, that post articles about web design and web technology. In one of their articles, [Schenker \(2016\)](#) writes about how to design good notifications. Schenker explains that notifications should be as unobtrusive as possible. They should not interfere with what the user needs to accomplish at any time, but still need to accomplish their intended purpose of notifying the user of important information. According to Schenker, the biggest job for the designer is to decide what notification should ultimately find its way to the user. Not all information is essential. He explains that notifications should be pushed if it has a direct impact on the user's UX, otherwise it probably does not need to be shown.

With all of this in mind, we figured out that we mostly wanted to make use of passive push notifications. To prevent the user from feeling overloaded with notifications, we needed to figure out what information would be essential. We agreed that information regarding bids would be crucial for both a transporter and an employer. We discussed whether or not notifications should be sent every time a new bid was submitted. We agreed that this would be too much, and could easily be experienced as annoying. To prevent this we came up with a solution where, by default, a transporter should only be notified when a new bid that is lower than the transporter's last bid has been submitted. Similarly, an employer should, by default, only be notified when a lower bid has been submitted. However, we did want to give the user a lot of control by giving the user the option to alter some settings regarding notifications. This would include the option to turn off notifications, the type of notifications that should be sent, and how often.

5.3 Considered functionality

In addition to the features and functionality we have talked about, we also had a number of ideas that we would have liked to do something with. However, due to the limitations in regards to time and resources for this project, we chose to not go further than the idea stage. In this section we will talk about these.

5.3.1 Complaints

Whilst we were certain that a profile page and notifications system should be a part of Backload, we also discussed other features that we thought could potentially be added into Backload in time. The possibility for users to leave complaints was one of these.

What we had in mind was a that for each transportation assignment, a transporter or employer could leave a complaint about the other party if they were unsatisfied. We were thinking that complaints should affect a user's reputation negatively. A user that has received a lot of complaints would receive some sort of punishment. What we were thinking here was to show an icon next to user's name to indicate that the user is not trustworthy. We did not go much further with this idea because we saw that it would be a time-consuming task to figure out the functionality and details around such a system.

5.3.2 Contracts and payment

We did briefly look into the possibility of allowing an employer and a transporter to create and sign a contract directly in Backload instead of having to do this internally between themselves. We were also thinking that payment from the employer to the transporter could happen directly via Backload. The main reason for implementing such a functionality would be to make things easier for the users by not having to use multiple systems. If Backload were capable of handling everything from finding a transportation assignment, to creating and signing contracts and then finally payment, there would be no need to use other systems. However, we decided to not pursue this idea. Just like the idea for complaints, we thought this would be too time-consuming. In addition to this we also saw a big security risk due to our lacking skill set in the area of creating systems that use electronic payment.

5.3.3 Notes

We discussed the possibility of adding a note to a transportation assignment. The main functionality of this note would be to store information the system itself does not store. For example you could write which truck you planned should take the transportation assignment on which route. This would be helpful for remembering if you have are keeping track of a lot of transportation assignments. It would also help communicate this information to your colleagues who might have to take over a bidding war, and needs the original planners intentions with the transportation assignment.

5.4 Layout

After establishing the necessary elements for each page in the application, the next step was to decide on placement and dimensions for the elements. The most time-consuming and difficult part of the layout was by far the main page containing the map and the table, which was a heavily discussed topic. To find the right balance between map and table we had to analyze the end-users needs so we could understand the importance of each element in the user experience.

We figured that the end-users would use both the map and the table a lot and the end users probably would use the application in different ways. By running some scenarios on the other members we found that:

- 1: The table is the most important aspect and some users would use only the table if they don't need a map.
- 2: The input and filtration must be easy to access, but when not used it should be possible to hide it to give the table and map bigger space.
- 3: The map should be as square as possible, and we should not turn it (North must be up).
- 4: The end user should be able to customize the application to suit their focus.
- 5: Navigation must be fast and easy to use.
- 6: Bidding is information not needed if you are not interested and should be hidden, but must be easy to find and access.

The elements needed on the main page is a map, a table, navigation, bidding, and filtration. We soon found that this was a hard task. A normal computer screen is a bit wider than it is high, so that is what we started designing for. We soon realized that the table had to contain a lot of information because the decisions are taken quickly based on information, so the information must be available. This resulted in a table that is very wide, and the height of course depends on how many elements the table contains. Each row in the table would also contain a button for showing the route on the map since that feature should be very easy to find and use.

The first step we did was following the norm and place the navigation on top where it is easy to access. Since the navigation must be fast and easy to use we also found that it would be a good idea to have it stick to the top of the browser if the user scrolls away from it so that it is still available. Now the big problem became really apparent, how can we with a wide table and a square map make something that is sensible and functional? The map can not be placed at the side of the table without sacrificing or hiding information in the table, and if placed below or above the table it would be very wide but not tall, not exactly the square we wanted. At this point we understood that we really had to use all

the space available to us, we could not afford all the space we wanted. Not only did we face this problem, but we also needed somewhere to put the filtration and the bidding. At this point we divided the main area in two with a horizontal line in the middle, and since it would be a much better mental model that the rest of the table was located below the edge of the screen, not behind the map, we placed the table at the bottom and the map on the top. See figure 5. This was only a temporary solution since we now had a map that was very wide. Doing this we could try out how the filtration and bidding would impact the map and table.

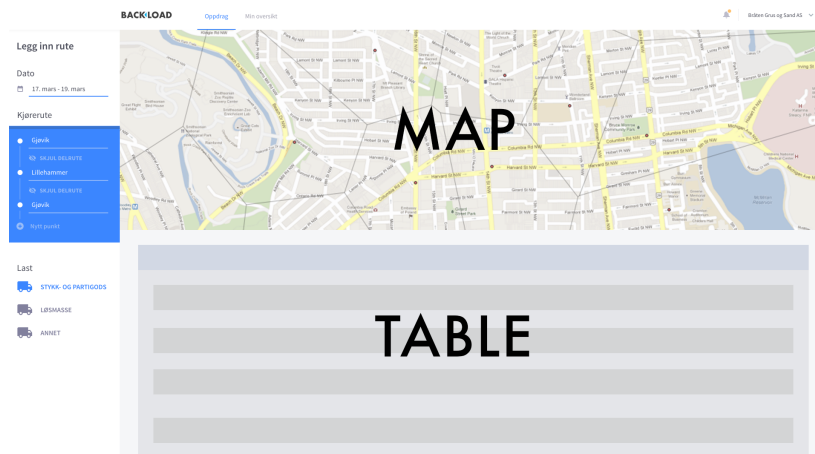


Figure 5: Our initial layout

Since we saw no obvious solution to the table/map problem we decided to attack the problem from a new angle. The filtration and the bidding needed space somewhere, and if we figured out where, it would possibly be easier to solve the table/map problem we were facing. Because we had decided that we needed to save space for the map and the table we started looking for ways to minimize the impact filtration and bidding would have on the rest of the elements. We started with the filtration.

The filtration could not be placed in the main area because that would not make sense so we had to put it in some edge of the screen or in the navigation bar on the top. We did not want to put it in the navigation since that would clutter it up and the space there should be reserved for eventual expanding of the navigation. Now we only saw one option, to place it along an edge. Looking on different sites we saw some solutions that was repeated all over, mainly filtration in a sidebar on the left side or putting it in the top or bottom of the main area. If we put it on the top or bottom of the page we would make the map and table, that we visualized taking most of the screen, have even less height. The height of the monitor is already smaller than the width, so this would make the map and table even more cramped in the height and very wide. The only sensible solution we could find was to put it in a sidebar on the left side. We think this is a good solution because a sidebar could also be hidden, saving space for the more important

elements, and if opened it would at the moment make the map more square and since it is not too big it would not make the table much more cramped. We did not think the table would take the entire width of the screen anyway, so a sidebar would not hurt the main elements.

The next thing we had to look at was the bidding. Every item in the table needed separate information for bidding, and with the table already very wide we could not put it in the table rows, but we did not like that idea anyway since we imagined it would look weird. Because it is a lot of information and the end user probably is not interested in the bids for each and every assignment, we decided that the information should be hidden somewhere. The big problem now was to find a sensible and logical way for the end user to open it, and find space for showing the information. The most logical place for the end user to interact is the rows in the table, the assignments themselves, and get more information and bidding by clicking on them. After deciding the most logical way of finding the information, we now had to find a good place to show the information without hiding the relevant information for that assignment, the table row and most likely the map. The only thing not interesting when looking at an assignment is the other assignments, so we decided that we would show bidding and possibly more information in an expandable row or new panel below the row.

At this point we also met a dead end. Someone mentioned that instead of each row having its own panel for bidding we could have one that is dynamically updated. We also had the problem with the map much wider than the square we wanted. So in a stroke of genius we tested how the solution would look if we chopped the map in half and put the bidding to the left of the map. See Figure 6. It looked decent, and the map was much more similar to a square than before. Problem solved, we thought. But not long after we realized that now the map was really tiny and not usable at all, and when we tried to make it a bit wider it looked weird since there were no symmetry in the design. We ended up scrapping this idea and went back to the original. Another alternative was doing what we found Google Maps to be doing, using the map as a fixed background and everything else as overlays. This was however quickly voted down since the table should have the main focus and we feared that in this solution the map would have the main focus. See Figure 7.

The current solution was a navigation bar on the top, sidebar featuring the filter, and a main area split in the middle with a map on the top and a table on the bottom. We did at this point not see any viable solution to the problem with the map, so we tried something new. Splitting the map and the table vertically instead of horizontally. See Figure 8. But this resulted in a map stretched the other way and a very compact table that became almost unusable.

Further we found that using only the lower half for the table would be bad for the end user, when browsing for assignments the end user would only look at the table and

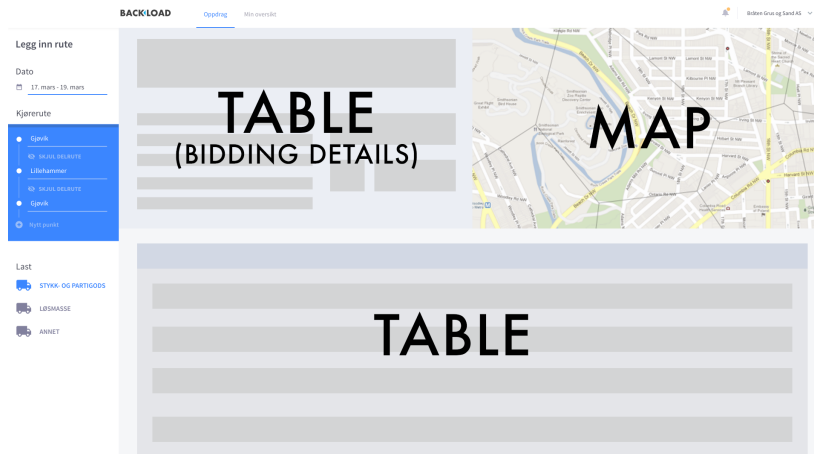


Figure 6: Main view split into 3 parts

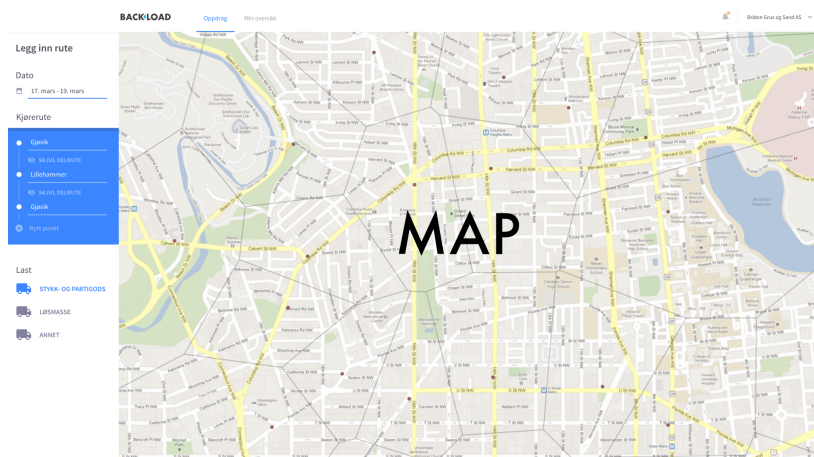


Figure 7: Map as background, sidebar and table as overlay. Table hidden.

not the map. Because of this we made two views, one with the split and one where the table was pushed up and covered the map. Switching between the modes would be done by pressing a button on the split to change mode. We also thought about making a third view for only the map, but we found no scenario where that would be useful to the end user even if it solved our problem with the stretched map.

After discussing the matter even further we agreed that this was one of the hardest tasks we had faced, and that the problem was simply like a puzzle with pieces that did not fit together. An unconventional solution would be a draggable split, but most of the group did not like draggable splits in websites because it is not common and not intuitive enough so we left the idea. Not finding a solution and after schedule we could not afford using much more time on this problem, so we had to leave it as it was. We argued that if we made the map smaller it would be usable, but if we left it wide it would be even more usable, but just in the east-west direction.

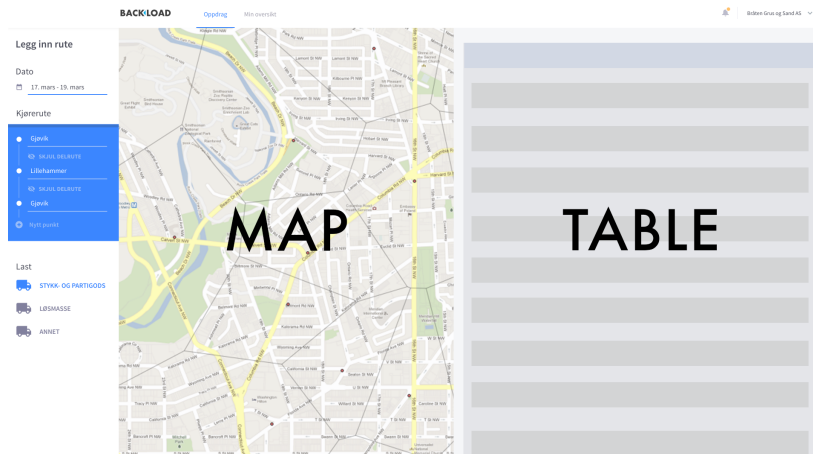


Figure 8: Map and table side by side

Overview

We knew we wanted a quick way for the user to find back to the assignments he had gotten, the current bidding wars, saved assignments and the old assignments that were completed if some information were needed. This we wanted to place on another page, and since they are related in a way we wanted to keep them close, but not in the same view since that would contain a lot of useless information if you were looking for something specific. We decided to link to the new overview page in the navigation so that it would be easy to find. Since we also had used a sidebar in the main page it would make sense to also use one here, and that would make it easy to switch between the different kinds of information. The sidebar would now contain four different categories of information the user could switch between. The assignments the user have gotten should contain all the information from the table but there is at this point no real need for a table so it can now be presented to the user as a card or something similar. The active bidding wars should contain the same information as from the table, but it must also feature the possibility of bidding and an overview of the bids. We think that there possibly would be a lot of saved assignments since some probably would save every interesting assignment, so this should probably be a table. The archive or history containing all the assignments the user have completed should be saved, probably in a table.

It is hard to say how to present this data because a table may sometimes be needed and other times a table might be bad if there is little data. We do however not like it switching eg. from cards to a table if there is over 10 sets of data, this would be confusing.

5.5 Final Prototype

Up until we started the digital prototyping process, we had just sketched our ideas on paper. We had a lot of different examples to build on. What we quickly found out is that we still had things we needed to discuss. The majority of the time we spent working on

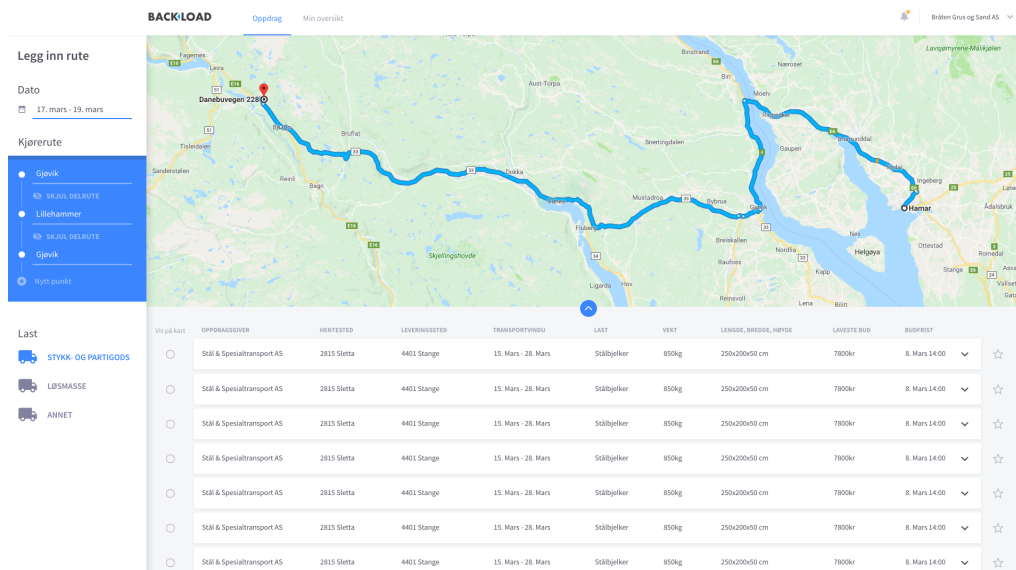


Figure 9: The main page of the prototype after date and route input

the prototypes went into discussing how the system itself should function. An example of this is how bidding wars should work. We landed on an iteration of this early in the process, but as we were working on the UI problems came up, and we had to rethink our approach.

For our digital prototyping tool we used Adobe Experience Design (Adobe XD). XD is an all-in-one prototyping tool for both user experience and user interface. You can swap between the static layout where you are designing and interactive prototype using these layouts. It also has share functionality to easily get feedback from users or colleagues(Adobe XD CC). Here you can find a online version of our [digital prototype](#).

The main focus of our prototype and solution in general is on the transportation assignment page. This is what the majority of the work went into. This page has the crucial functionality in our solution, and is supplemented by the rest of the pages. Some of these other pages are crucial for the application to do its job, like the bidding war page for example, but most of our focus went to the main page, where we saw the most room for making a good user experience.

When you first open the prototype you see empty input fields on the sidebar to the left, and a full table of all the transportation assignments. When you click on the input fields to fill them you get to the page shown in figure 9, which shows your route on a map at the top, and the transportation assignments along the route in the table below. From here you can click to open one table element to see more information and have the possibility to bid. You can also click the arrow and move the table up, so you only see table.

All the time you can navigate to the Overview page using the navigation bar at the

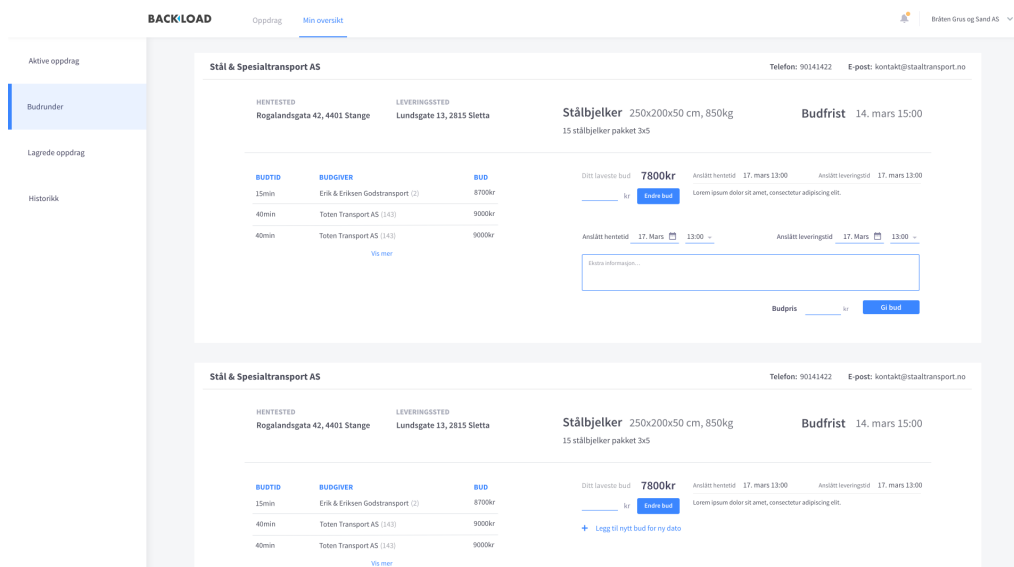


Figure 10: Ongoing bidding war on Overview page, input for adding new bid with separate date

top of the page. On the overview page you can swap between three different views, my active transportation assignments, bidding wars, and favorites. Active transportation assignments and favorites are static pages showing our intended design. The ongoing bidding wars page has the option to input a new separate bid with a new date (Figure 10).

5.6 User testing

When we had created a digital prototype we moved on to testing it, this test was conducted on fellow students on campus. We conducted our tests in a small room for groups of maximum eight people. From our group we had three people conducting the test. One was the conductor who presented the group and guided the tester through the test. One was an observer who looked at how the tester interacted with the prototype to try and learn something from that. The last group member took notes from what the tester said about the solution, both during and after the test.

We did six tests in total, one of these was with two testers together. We did the duo test so they could discuss and bounce ideas off each other, to get some new insight into what people assume when they look at the application. Even if you ask the user to *think out loud* most people will not be able to do this fully. Therefore, if the two testers are discussing something, they are forced to say it out loud, and we can record the insights they provide. We prepared eight tasks (Appendix I) for our testers to complete, and took notes for each task individually.

5.6.1 Result and discussions

We saw some trends emerge from the test results, showing us how certain functionality in the application does not work. A good example of this is the hide sub-route button. One problem that occurred with this button was that the contrast was too small, another problem was that the name was not enough for everyone to understand what the button did. When we were designing the visuals around the button and the input for route, we had problems showing it properly without making it cluttered, so this is something we were aware of and were looking for feedback on.

Multiple testers gave feedback regarding the split between the map and people. Instead of using a button to control the amount of table that should be visible, they suggested to use a drag functionality. They would like to have greater control of how much of the table and map should be visible.

One big trend that most of the testers struggled with was adding a new separate bid on a transportation assignment. Most people tried to do it on the transportation assignment page, but in our prototype it was only possible on the Overview page. This is an error we would have changed at once in a new iteration of the prototype, there is little to no reason to not have the possibility both places.

Some people got confused about the wording active transportation assignments, and thought that it meant active bidding wars. Most of them realized the difference when they saw the other tab called bidding wars, but the name should be clear enough by itself.

Feedback from supervisor

We also had a thorough discussion with our supervisor after we had created the prototype, and conducted the user testing. He made us aware of the fact that the solution we had come up with had a lot of wasted space in the map view, that showed parts of the map that was not relevant. For example, when plotting a vertical route on the map, then large portions on each side of the route would not show anything relevant. You could easily end up with a big chunk of ocean to the side of the route, which is not particularly useful for the user to see. This is space that we could have used in other ways in order to give more space to the table with transportation assignments.

One potential solution for this could be to allow the user to choose between a vertical split between the table and the map, instead of just the horizontal split we have chosen to go with. This could work well for routes that are either vertically or horizontally straight, since the user can switch between the two options for the split between table and map. However, this would not solve the situations where a route is plotted in such a way that the path of the route moves both horizontally and vertically. For example, where the first half of the route is pretty much a straight horizontal path, and where the other half is more of a vertical path. A route like this, that would force the map to be quadratic would

happen less frequently than just a vertical route, but it would still be a big issue.

Another solution that might work is to allow the user to show certain elements of the application on multiple screens. From what we learned from our research and interviews of the target audience, we knew that it is normal for the people working at the office to have multiple computer screens. With this in mind, we thought of a solution where the user could view the map by itself on one screen, and view the table by itself on another screen. This would allow for the map to be displayed in a much more optimal quadratic form that can account for all kinds of routes. In addition to this, it would give us much more space to work with for the display of the transportation assignments.

At this point in the process, we felt like we did not have any time left to make any drastic changes to the design of the solution if we were to get any implementation done. This is why we decided to not go further with any of these two solutions. However, we did want to take some measures to account for the problem. Earlier on, we had discussed using a drag functionality in order to control what portion of the map should be visible. This was something that we had gone away from since most of the group members did not like this functionality. However, since we received feedback during the user testing that this could be beneficial, we decided to implement this functionality anyway. We are aware that this is not the most optimal solution for the problem.

6 Implementation

We will in this chapter introduce the technologies we used and show the process we went through when implementing.

6.1 Technologies

HTML, CSS and JavaScript

We have used the basic languages the browser understands to make the project. This includes HTML5, CSS3 and ECMAScript 6. The HTML will be rendered by Angular, the CSS and JS will be the result of angular-cli compiling SASS and TypeScript ([HTML and CSS](#)).

Sass

Sass is an extension of CSS that adds some new features that does not work in original CSS. These features are things like variables, mixins, nested rules and inline imports. There are at the moment two syntaxes for Sass, the original and old Sass, and the new SCSS that is more similar to CSS. We have chosen to use the original Sass style syntax since we think it is a lot cleaner to write since we do not have to use the curly braces and semicolons like in CSS and SCSS ([Sass](#)).

TypeScript

TypeScript is a superset of JavaScript developed by Microsoft that compiles to normal JavaScript and is also included in Angular. It gives some new features like types that results in less errors at runtime if used properly. As a superset, normal JavaScript is also valid TypeScript ([TypeScript](#)).

Node.js

To develop the application locally we have used node.js. Node.js is built on Chrome's JavaScript engine, this makes it lightweight and efficient. This is command-line interface. We have used it only for the Node Package Manager ([Node.js](#)).

NPM

The Node Package Manager is a package manager that let us grab packages from external sources. This makes it a lot easier to bundle together the code we need from different open source projects to give us the base of the application. NPM will also keep the code up to date ([NPM](#)).

6.2 Angular

Angular is a front-end framework developed by Google. One of our goals was to learn something new. Some of us have had some experience with the older AngularJS framework, but the new Angular was something new to all of us. One of the things we figured out quite early was that we wanted a single page application that loads everything dynamically, we think this is the future of the web because it is a smoother user experience than the traditional loading and flashing. To accomplish this we decided we should use a framework. The frameworks we thought about using were Angular, React and Vue. We made the decision to use Angular based on the documentation and modules offered, and the use of TypeScript that we wanted to learn. We also all like Google and believe they make solid software so this probably influenced our decision ([Angular](#)).

Angular is made up of different building blocks. Components are views containing sets of elements. Every application made in Angular must have at least one component, the root component that connect the components to the DOM. Each component defines a class that contain application data and logic and is connected to a HTML-template. A component can contain other components, and the different views are made by implementing logic showing different components.

Services are classes used by components to access data and logic that is shared or simply just not associated with a specific view. They work by injecting them into components as dependencies, that means that the component is depending on it to work as it should.

By dividing the structure into components and services the structure gets cleaner, and easier to manage since the tasks are divided.

There is also data-binding, routing and other blocks. Angular even let you modify components before they are shown by using template directives. Since this product is only a prototype we did not use the full power of Angular and stayed mainly in the component and service area using data-binding to show data in the templates. ([Angular Architecture](#)).

Angular-cli

Angular-cli is a command-line interface tool that have a lot of great features built in. By some short commands it will generate new files, compile files and even host a local webpack server that show the current project in the browser, and update it every time we save a file ([Angular CLI](#)).

6.3 Google Maps

Google Maps is a web-based mapping service that provides detailed geographical information worldwide. In addition to the interactive digital map, it offers a comprehensive list of features like route planning, real-time traffic monitoring and street view ([Google](#)

[Maps Platform](#)).

To implement Google Maps in the application we used Angular Google Maps, which is a package that provides Angular integration for the core functionality of the Google Maps API ([Angular Google Maps \(AGM\)](#)). Although Angular Google Maps is built on the Google Maps API, it is not created or supported by Google. Therefore, a lot of the functionality you would find in the official Google Maps API is not included.

Using the official Google Maps JavaScript API would also technically be possible, but then we would not be able to utilize all the features offered by Angular, and most likely we would encounter import problems. Which is why we ultimately decided to use Angular Google Maps, as it is a more reliable solution.

Visualizing routes on the map is achieved by using [Agm-Direction](#), a directive for Angular Google Maps that integrates the direction service from the Google Maps API ([Agm-Direction](#)). This is the part of the Google Maps API that calculates directions between locations and builds a route that can be drawn on the map.

6.4 Material

Angular Material is a versatile and consistent package for Angular that comes with functional and styled components ready to use. The components are made for Angular, by the people that make Angular. They are also styled in the typical flat design made by Google called material design ([Angular Material](#)).

The following components from Angular Material are used ([Angular Material Components](#)):

Datepicker: The datepicker is used in the filtration to let the user input a date.

Menu: The drop-down menu in the top is used to declutter the top menu.

Sidebar: This component is used to make the left sidebar containing the input-fields.

Toolbar: The menubar on the top is a toolbar from Angular Material.

Icon: The icons in the application is its own components.

Table: The table is a component from Angular Material.

Sort header: The sorting in the table is a component.

Form Field: Wrapper for input elements.

Input: Input elements for data.

Divider: Divider line used in the navigation bar.

6.5 Drag slider

This is a third party UI library that allows to re-size elements using a split-bar. This is not a package made by the official Angular team. We had to use this because there is no official Angular module that let you do this. The slider comes with a lot of different values that can be adjusted and customized when needed ([Angular Split](#)).

6.6 Backend

This application needed a solid back-end to work since the front-end is in many ways just a shell presenting data. To make this we needed a way to store data and a way to transfer data. The way we have learned to do that in our field of study is by using SQL and PHP.

6.6.1 Database

Our database structure has seven tables total, most of these are small tables with 3-5 columns but the assignments table is large with 22 columns(Figure 11). This table holds most of the data that goes into the table as well as the data used for plotting the transportation assignments on the map. When creating the database we first created a very simple sketch of an ER, for more of a frame of reference while we discussed possibilities. When working on the ER diagram stage we tend to think ahead to the relational model, therefor we prefer working from the relational model instead of an ER diagram, and discussed more here.

There are some aspects of the database that were up for discussion. For example should we store users in two different tables, or in the same table with a token to identify which type of users they are. The main argument for storing them in the same table is that they both store the same data, so it would be two identical tables apart from name.

For implementing the database, we created a setup.php file that deletes the database if it already exists, and sets it up, with the dummy data we have inserted into setup.php. This is to make it easier if we have to make small tweaks to the database and everyone else can just run setup.php again and the database is up to date. For deploying on a server the database information in setup.php would have to be changed for the deployment server. At that point one could also just use the same SQL from the setup file in a interface of the servers database.

6.6.2 PHP

PHP is a programming language that can be used for many things. It is used a lot in web development. The language can do a lot of different things and it is currently used to power a lot of dynamic things on the web. Even WordPress is built in PHP. In this project we used PHP to connect the front-end to the database and relay data both ways ([Hypertext Preprocessor](#)).

Table Name	Columns and Datatypes
users	<ul style="list-style-type: none"> user_id INT name VARCHAR(100) email VARCHAR(50) password VARCHAR(256) phone_number INT(11) zip_code INT(4) city VARCHAR(50) address VARCHAR(100) token INT(1)
saved_assignments	<ul style="list-style-type: none"> user_id INT assignment_id INT
active_assignments	<ul style="list-style-type: none"> user_id INT assignment_id INT archive_status TINYINT
notifications	<ul style="list-style-type: none"> notification_id INT assignment_id INT bid_id INT read_status TINYINT token INT(1)
bids	<ul style="list-style-type: none"> bid_id INT suggested_start DATETIME suggested_end DATETIME message VARCHAR(256) user_id INT assignment_id INT
bid_amount	<ul style="list-style-type: none"> bid_id INT amount INT(10) time_given DATETIME
assignments	<ul style="list-style-type: none"> assignment_id INT type INT(1) window_start DATE window_end DATE start_zip_code INT(4) start_city VARCHAR(50) start_address VARCHAR(100) start_lat FLOAT start_lng FLOAT end_zip_code INT(4) end_city VARCHAR(50) end_address VARCHAR(100) end_lat FLOAT end_lng FLOAT bid_deadline DATETIME short_desc VARCHAR(100) description VARCHAR(256) weight INT(15) height INT(10) width INT(10) length INT(10) user_id INT

Figure 11: Our database tables, with all columns and datatypes

6.7 Developing application

6.7.1 First steps

The first step in implementing the functionality was setting up a repository in GitHub. The repository was made private since the group and Johannes agreed that it was no reason to make the idea public. After initializing the repository everyone got invited to the repository and got full access.

We had to set up node and npm to get the packages we wanted to build the project. Node is downloaded from the website and installed like any other computer program. Included in the node install is also the node package manager. To check that we had gotten the latest version of npm in the installer we checked the installer for the version

number using the commands "node -v" and "npm -v" in the terminal. Npm is updated more frequently than node, so we used the command "npm install npm@latest -g" to update npm to the latest version.

The group did at this stage have a repository working and the basic terminal tools installed. The next step in the process is to install the framework Angular itself. We wanted all the tools the angular-cli interface gives us so we installed the angular-cli package using the node package manager using the terminal command "npm install -g @angular/cli".

We used what at the time was the latest stable version of Angular, version 5.2.9 of Angular. To set up an Angular project the command "ng new my-app" is used, there the my-app part is the name of the application. Since we wanted to use Sass instead of CSS, we searched the internet for a way to do this. Luckily this was already included in the angular-cli interface, so we could run the command "ng new Backload --style=sass" command to get the project up and running, and also using the Sass stylesheets without having to manually configure too much. The next step was making things easier for ourselves when developing.

At this stage everything is ready for developing, so now the members started looking at different things. Since the googlemaps-API was critical we placed two of the members there. One member started working on the database, and the final member started setting up the structure of the application.

To help make the development a bit easier and quicker, we made use of the Angular Material package that offers some components that are ready to use. For the layout of the page we used the Sidenav and Toolbar components. The Toolbar component has been used to create the top menu. The Sidenav component was used to create the sidebar to the left. Due to how this component works, it is also a wrapper for everything else, which means that both the map and table is inside this wrapper.

We encountered a problem when we were trying to make the draggable slider between the map and table. Unfortunately we could not find an official package for this. However, after searching for quite some time and testing a few different solutions, we found a package that we could use. The package called angular-split would let us do exactly what we needed.

6.7.2 Folder structure

In order to make things easier for ourselves when coding, we spent some time figuring out how to set up our folder structure for the project. In addition to the files that is automatically added by angular-cli when creating the project, we have added our folders and files into the source files of the project. This is located inside the src folder. Inside the src folder, angular-cli automatically creates the app, assets and environments folders. Figure 12 shows the folder structure we ended up with for the source files.

Inside the app folder, we have created three folders: components, services and shared. We expected to create a considerable amount of components and services. We felt keeping the code for these separated inside their own folders would make it more clear and faster to get back to the code for a certain file that has anything to do with a component or service. We ended up with quite a few components, and Figure 7 shows the complete structure for the components folder. The shared folder is meant as a folder to put any code that does not belong to a specific component or service. It is code that can be used in multiple components and services.

The data folder was used throughout the development process as a folder to store JSON files. These files were used in the early stages when the database and PHP had not yet been properly implemented.

We have structured the sass folder in such a way that it would be easier to keep the code organized by using sub-folders that are specific to certain areas of the application. See Appendix O for a full overview of the Sass structure.

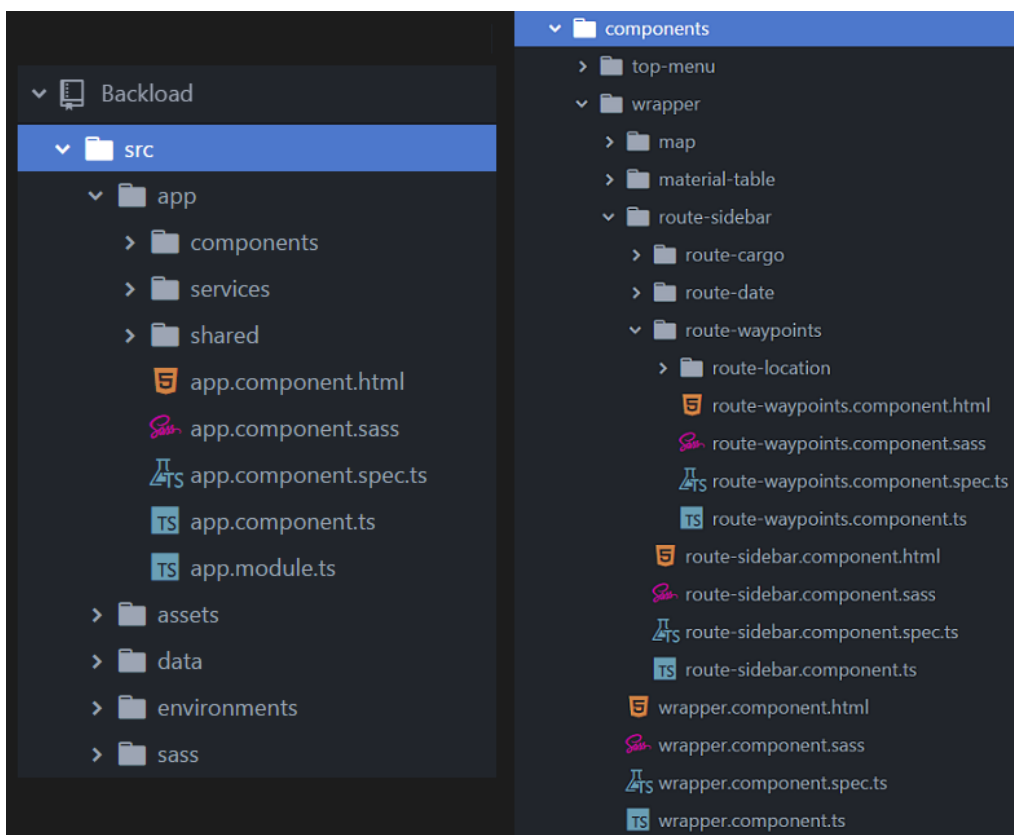


Figure 12: The left side shows the source files for the Angular project. The right side shows the structure for our components

6.7.3 Table

We have created a component called `MaterialTableComponent` that handles the retrieval of data for all transportation assignments from the database and displays them in a table. We have made use of the Angular Material Table component, which provides a Material Design styles data-table ([Angular Material Table](#)).

Retrieving data from the database

We have created a back-end PHP file called `deliveryJobs.php`. In this PHP file, we retrieve data about all of the transportation assignments and the bids related to each one, and then returns this data.

In the TypeScript file of the `MaterialTableComponent`, we have created a class called `JobListingsDatabase`. See appendix [P.1](#) for the code used in `JobListingsDatabase`. Here, we make use of a service called `DatabaseService`. This service contains a method called `fetchDeliveryJobs()` that simply sends an HTTP POST request to the back-end PHP file. We call this method inside the `JobListingsDatabase` class to retrieve the data that is going to be used in the table.

Populating the table with data

In the TypeScript file, we have also created a class called `JobListingsDataSource`. In the `ngOnInit()` method, we have set the variable `dataSource` to be an object of this class. The `JobListingsDataSource` class returns the data to be used in the table. When any changes are made to the data, these changes will also be updated in the table. The class also contains a method called `getSortedData()` which handles all of the sorting. This method is called every time a column header in the table, that has sort-functionality, has been clicked. The data returned from the `JobListingsDataSource` will then change, causing the table to display these new changes based on the sort. See appendix [P.2](#) for the full code used in the `JobListingsDataSource`.

Stack Overflow refers to itself as the world's largest developer community ([Stack Overflow](#)). When doing the implementation of the `JobListingsDatabase` and `JobListingsDataSource`, we have used a specific discussion on Stack Overflow to a large extent. The `getSortedData()` inside `JobListingsDataSource` has been directly copied and slightly altered in order to fit our needs ([Angular 2 material implementing sort, filter and pagination](#)).

After the table has been populated with the data from `dataSource`, we have added the HTML used to show the expanded content that appears when clicking on a table row. This HTML is different for every row, so we have added this dynamically with the correct data by using a service called `TableService`. We have made use of a method that Angular provides called `ngAfterViewInit()`, which is automatically called after a component's view has been fully initialized. Inside `ngAfterViewInit()` we have looped through all of the rows and called the `createHtmlForExpandedRow(job)` with the row as parameter in order to

show the correct data for each row's expanded content. When a table row is clicked, the function `onRowClicked(e)` is called. This will either show or hide the expanded content row, depending on if the expanded content is already shown or not.

Using the `<mat-table>` component

The Angular Material Table component is accessed by using the `mat-table` tag in the HTML file of the `MaterialTableComponent` we have created. As you can see in Listing 1, the `mat-table` takes several attributes. Data is provided to the `mat-table` via the `[dataSource]` attribute. This is directly linked to an array in the TypeScript file of the `MaterialTableComponent` containing objects representing each transportation assignment. The table renders a row for each object in the data array. The `matSort` attribute is provided for the `mat-table` in order to make use of its sorting functionality. The `matSortActive` attribute is used to decide which column of the table that should be sorted as default. We have set this to be equal to `"transportationWindow"`, which sorts by start date of the transportation window. We have used the `matSortDirection` attribute to decide that the data should be sorted in ascending order.

```
1 <mat-table id="table" #table [dataSource]="dataSource" matSort
2   matSortActive="transportationWindow" matSortDirection="asc">
```

Listing 1: `<mat-table>` tag used in HTML to create a Material table

6.7.4 Map and routes

Retrieving data for locations and routes, and rendering map, map-markers and routes is managed by the `MapComponent`. The `MapComponent` is a custom component we created, containing all necessary components imported from installed packages. Angular Google Maps generates the map itself and map-markers, and `agm-direction` creates and displays routes.

Centering map on user geolocation

When the map is initialized the function `setCurrentPosition()` gets called to center the map on the user's geographical location as seen in Listing 2. This function takes advantage of the HTML Geolocation API, which provides a method of accessing user location in web applications through GPS or IP/WiFi location ([Geolocation API Specification](#)). It is supported by nearly all browsers, but as a privacy security measure the geolocation service requires user permission to obtain location data. When requesting location the browser will automatically display a dialog box that the user can either approve or deny.

```

1 private setCurrentPosition() {
2   if (navigator.geolocation) {
3     navigator.geolocation.getCurrentPosition((pos) => {
4       this.mapsService.setLatLngCoordinates(pos.coords.latitude, pos.coords.longitude);
5       this.zoom = 7;
6
7     }, (error) => {
8       this.showError(error.message);
9     }, { enableHighAccuracy: false, maximumAge: 15000, timeout: 5000 });
10  } else {
11    this.showError('Geolocation not supported');
12  }
13 }

```

Listing 2: Function utilizing HTML Geolocation API to obtain geographical location of user

Retrieving data for transportation assignments routes

The route coordinates for all transportation assignments is fetched through a service called MapService. When populating the JobListingsDatabase inside the MaterialTable-Component, the function setDeliveryJobRoutes(routesArr) is called. This function make use of the data retrieved from the database, and updates an observable containing coordinates for all the routes. The function also removes transportation assignments that are filtered out by calling another function within MapService called filterDeliveryJobRoutes(). By checking for hidden transportation assignments in the table, this function makes sure that the transportation assignments hidden from the table will also not be shown in the map.

Input data for user plotted route

In order to plot a route on the map, we have utilized the Google Maps Autocomplete feature included in the Google Maps API. This feature generates suggestions of places based on the contents in the route input fields located in the sidebar-menu.

When the user selects an item from the drop-down of place suggestions, the coordinates of the selected place will be sent to the MapService and added to the user-route object. When both origin and destination is set in the user-route object, the route path will be created and displayed on the map. If the user changes or adds more points to the

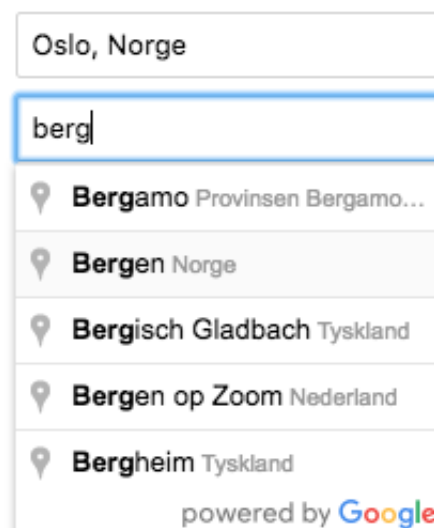


Figure 13: Example of Google Maps Autocomplete with a list of suggested places

route, the route on the map will get updated to match the changes.

6.7.5 Filtering transportation assignments by distance

Backload is based on the fundamental principle that finding transportation assignments should be quick and easy. An essential part in our efforts to achieve this is how only potentially suitable transportation assignments will be displayed to the user: transportation assignments that are unattainable or impractical will be filtered out, for instance those that deviate too far away from the users planned route. We implemented this into our solution based on a function from Stack Overflow that we customized to our situation ([Brown, 2013](#)).

Implementing the functionality for filtering transportation assignments by distance was one of the most technically challenging tasks in our development process. The premise of our approach is simple, for each transportation assignment we calculate the distance between pickup-point and the user's route, then check if it is within a determined max radius. The difficult part is finding the point in the route path closest to the pickup-point, which is where we need to calculate distance from. In order to do this we utilized a property of the Google Maps route object called "overview_path", which is an array of geographical coordinates that forms a simplified version of the entire route (See [Figure 14](#) for a visualized example).

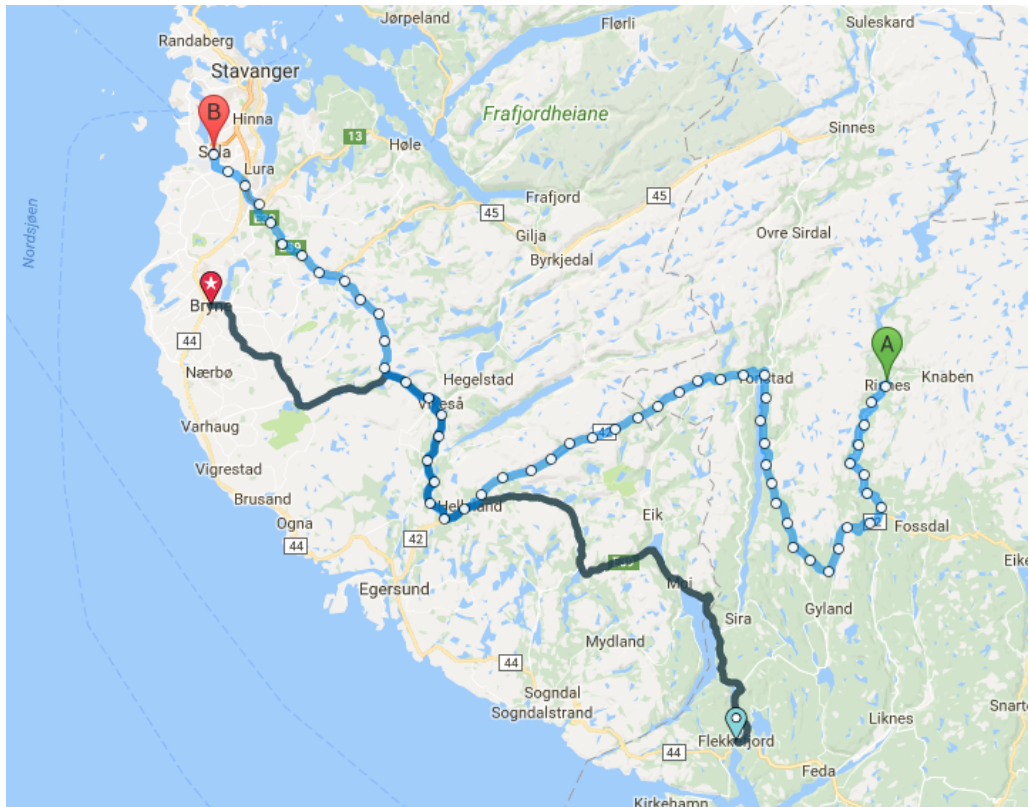


Figure 14: Visualization of the `overview_path` property in a route, the white circles represents the geographic points contained in the `overview_path` array

To find the point in the route closest to the pickup location we created a function that loops through all points in the route, then utilizing a function included in the Google Maps API we retrieve the aerial distance in meters to the pickup location for each point.

After calculating distance for each route point, we can identify the point closest to the pickup location by the point with the shortest distance (Figure 15). The transportation assignment will then get filtered out from map and table if the closest point is not within the maximum radius, and the operation will repeat for the next transportation assignment until every transportation assignment has been checked.

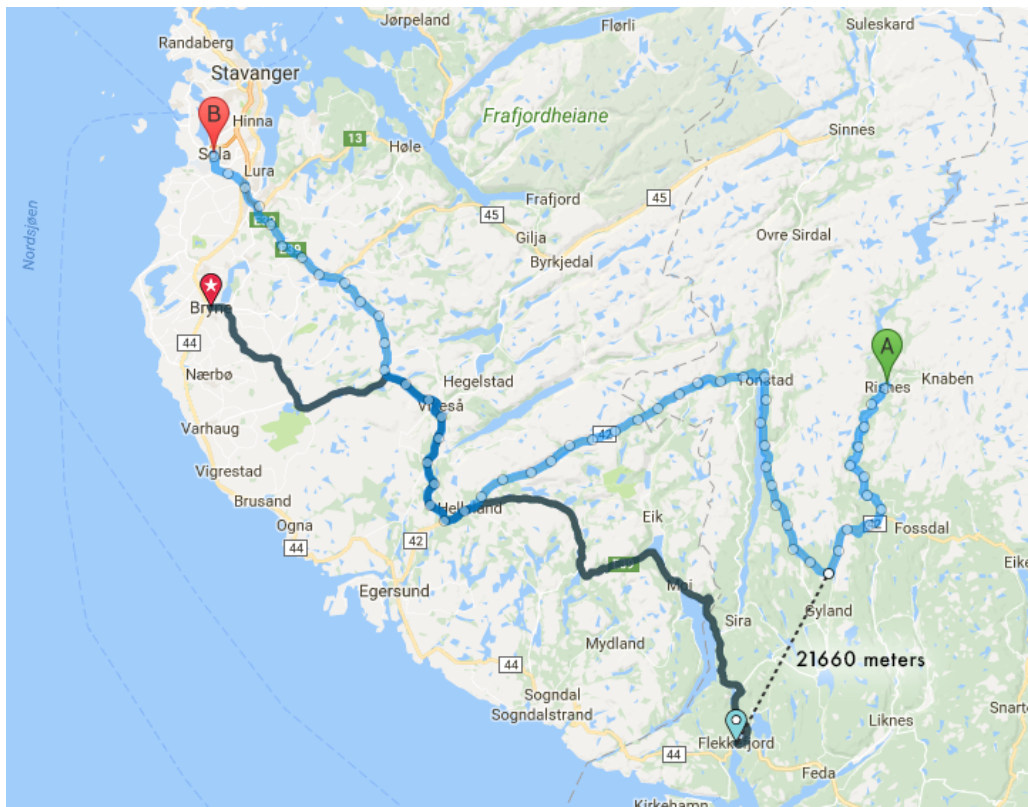


Figure 15: Visualization of how distance between route and pickup location is calculated from `overview_path`

6.7.6 Filtering by user input

The filtration is two date-pickers in the sidebar that impacts the data shown in the table. The components used for the date-pickers are from Angular Material. They are mostly ready except for the format that is American format. Since this is not important we did not use time to change it, it was surprisingly hard to configure. The filtering and the table are two different components, and they have no relationship. To make them "talk" we made a service for sharing data. The most important parts of it can be seen in Listing 3. By using a service every component that are connected to it will have the data it contains available to use, and every component can change the data if needed.

When the dates in the date-pickers are changed by the user of the application they are sent to the service. The dates in the service is then available for some JavaScript that updates the table. If a start date is sent the JavaScript removes everything in the table that starts before the date given, and the opposite for end date, if an end date is sent the script will remove everything with an end date later than the end date sent.

```
1 // Relevant part of service
2 @Injectable()
3 export class DateService {
4
5     // Observable for the startdate
6     private startdateSource = new BehaviorSubject<string>("Default startdate");
7     currentStartdate = this.startdateSource.asObservable();
8
9     //Observable for the enddate
10    private enddateSource = new BehaviorSubject<string>("Default enddate");
11    currentEnddate = this.enddateSource.asObservable();
12
13    constructor() { }
14
15    // Change startdate
16    changeStartdate(startdatemessage: string) {
17        this.startdateSource.next(startdatemessage);
18    }
19    // Change enddate
20    changeEnddate(enddatemessage: string) {
21        this.enddateSource.next(enddatemessage);
22    }
23 }
```

Listing 3: Service for sharing filtration data

6.8 Final result

The final solution is a single-page application that is capable of searching for a route and visualizing this on the map. The application retrieves all the transformation assignments stored in the database and displays them in a table. Filtration of these results is possible by submitting a start and/or end date. The table results can also be sorted via certain categories. More details about a transportation assignment as well as bid information is retrieved from the database and can be displayed by expanding the row when clicked. A deployed version of the final solution is available at www.aslakhle.no.

6.8.1 Walk-through of the application

Figure 16 shows how the application will look like when initially loading the page. The map is centered on the user's location, if the user has allowed the application to get his geographical position. The table displays all of the transportation assignments stored in the database.

The user can plot a route by typing into the input fields in the sidebar. We have implemented auto-complete for the input fields, so the user will get suggestions for locations while typing. When the first location has been submitted, the map will center on the coordinates of that location and add a marker. When a location for the second input field has been submitted, the map will display a path between the two locations on the map. The map is re-positioned and the zoom-value is adjusted to make sure that the whole route is shown on the map. Nearby routes (75000 meters or closer) will also be displayed on

OPPDRASSIVNER	HENTESTED	LEVERINGSSTED	TRANSPORTVINDU ↑	LAST	VEKT	LENGDE, BREDD, HØYDE	LAVESTE BUD	BUDFRIST
Entreprenør Lorentzen AS	Hunddalen	Notodden	10/6/18 - 25/6/18	Vaskemaskin, 1 pall	60kg	140x100x100cm		5/6/18 15:00
Gjøvik Transport AS	Hunddalen	Nyborg	11/6/18 - 17/6/18	4 tørketrommeleer, 4 paller	170kg	150x100x100cm	600	7/6/18 12:00
Lillehammer Transport AS	Hunddalen	Roa	14/6/18 - 22/6/18	Vaskemaskin, 1 pall	56kg	140x100x100cm	700	7/6/18 14:00
Gjøvik Transport AS	Gjøvik	Fagernes	16/6/18 - 24/6/18	30 Stålbjelker, pakket 6x5	950kg	400x80x60cm	700	13/6/18 14:00

Figure 16: The initial view of the page

the map. Figure 17 shows an example where a route with start point Oslo and end point Bergen has been plotted.

OPPDRASSIVNER	HENTESTED	LEVERINGSSTED	TRANSPORTVINDU ↑	LAST	VEKT	LENGDE, BREDD, HØYDE	LAVESTE BUD	BUDFRIST
Entreprenør Lorentzen AS	Hunddalen	Notodden	10/6/18 - 25/6/18	Vaskemaskin, 1 pall	60kg	140x100x100cm		5/6/18 15:00
Gjøvik Transport AS	Hunddalen	Nyborg	11/6/18 - 17/6/18	4 tørketrommeleer, 4 paller	170kg	150x100x100cm	600	7/6/18 12:00
Lillehammer Transport AS	Hunddalen	Roa	14/6/18 - 22/6/18	Vaskemaskin, 1 pall	56kg	140x100x100cm	700	7/6/18 14:00
Gjøvik Transport AS	Gjøvik	Fagernes	16/6/18 - 24/6/18	30 Stålbjelker, pakket 6x5	950kg	400x80x60cm	700	13/6/18 14:00

Figure 17: The route between Oslo and Bergen has been plotted

The application allows for multiple waypoints to be added. When both a start location and end location has been submitted, a button for adding a new point will appear below the input fields. Clicking on this button will cause another input field to appear. The user can then submit a location for the new destination. The map will re-position and adjust the zoom level. Figure 18 shows how the map looks like after submitting a new destination for Ålesund.

The table originally shows all of the transportation assignments (unaffected by the search), but the results can be filtered by start date and/or end date. The user can either manually enter a date by typing into the input field of a start/end date field or click on the down arrow inside the input field to make use of a Material styled date-picker. If typing manually enters the date, it has to be in the American date format (for example "06/11/18")

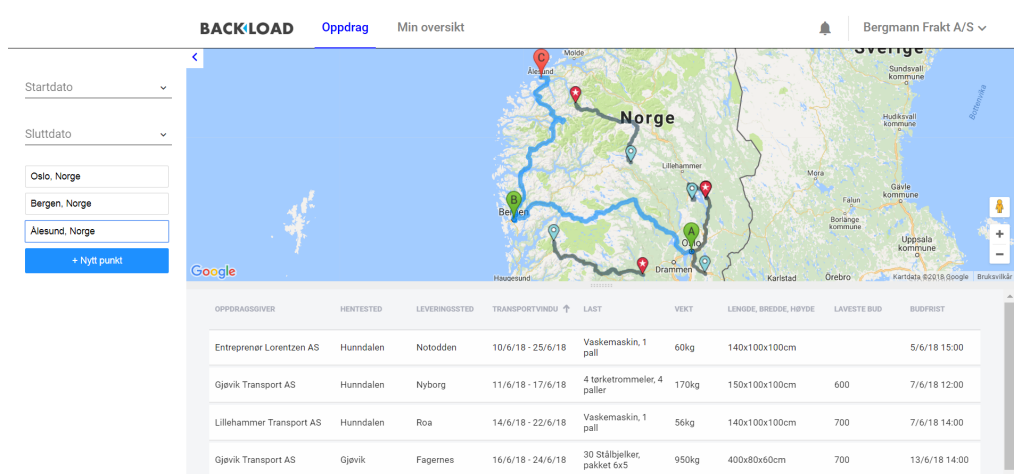


Figure 18: The route Oslo - Bergen - Ålesund

of month/date/year. If the user selects a date from the date-picker, this will be handled automatically. Figure 19 shows an example of where the table results has been filtered by both start and end date.

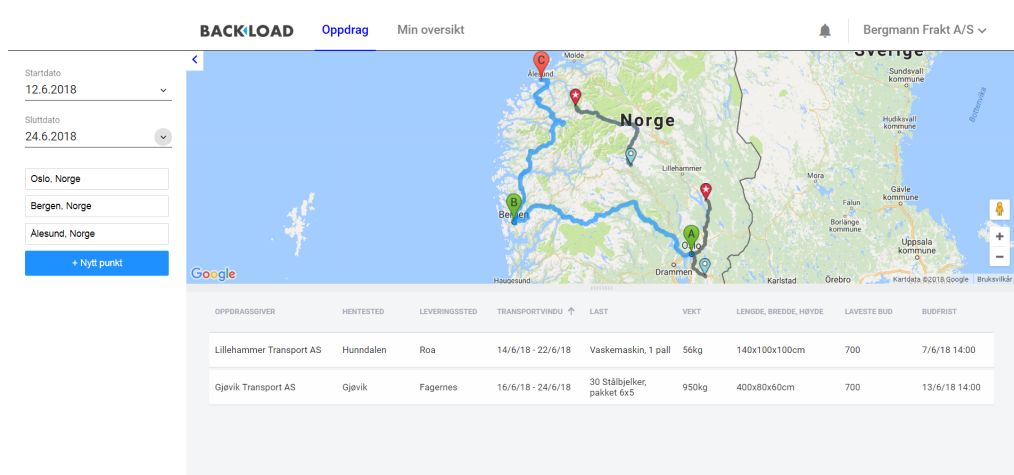


Figure 19: Shows the route Oslo - Bergen - Ålesund where start date and end date has been entered

The user can control how much of the table should be shown at any time by using the drag slider. It is possible to hide the map completely by dragging the table all the way to the top. Similarly, it is also possible to hide the table completely by dragging it all the way to the bottom. Figure 20 shows an example where the table has been dragged almost all the way to the top.

The user can view detailed information about a transportation assignment by clicking on the row. This will cause the row to expand in height, showing detailed information about the transportation assignment and the bids that have been submitted for that transportation assignment (if any has been submitted). The expanded row also allows shows non

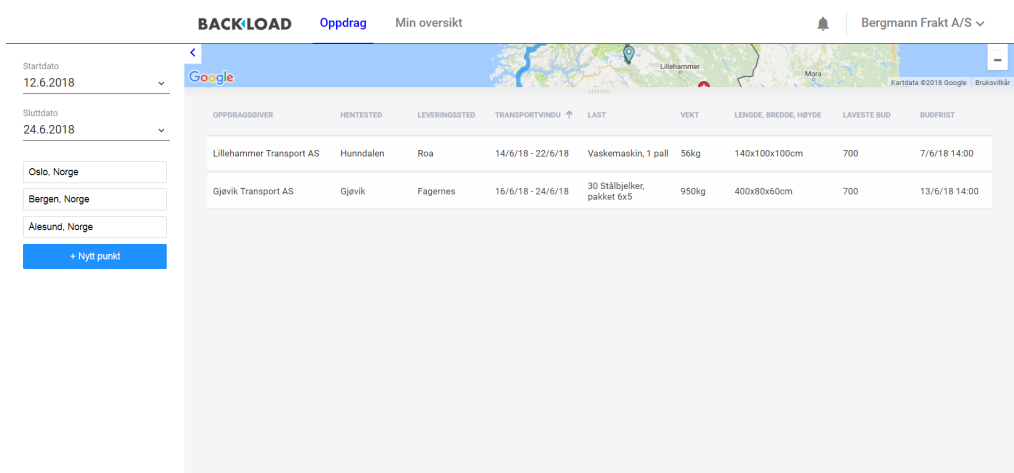


Figure 20: Shows how the drag slider has been used to give more room for the table working inputs for inserting a new bid.

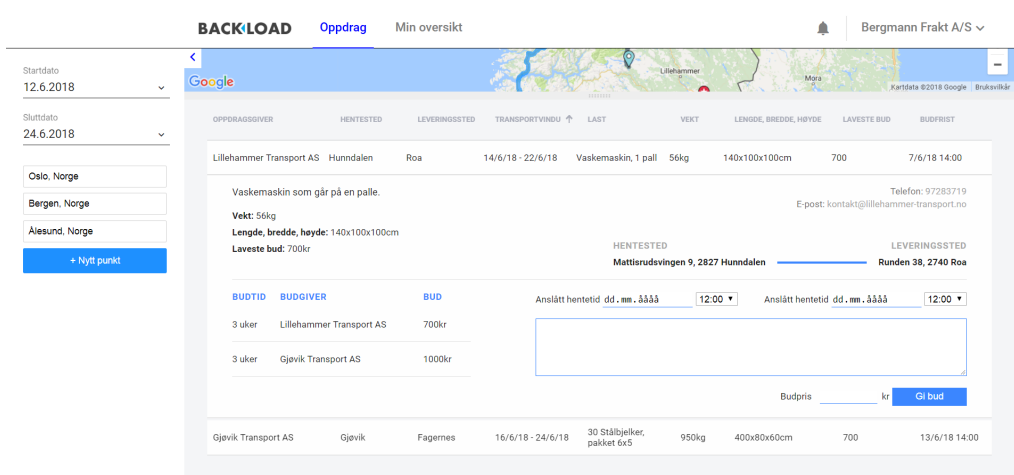


Figure 21: Shows an expanded table row

7 Process

In this chapter we describe and discuss our process throughout the project. First we define our roles, and tools we have used, then there is a discussion chapter where we look at our own process.

7.1 Role distribution

Håkon: Scrum master and group leader.

Christoffer: External communication.

Mats: Creative/design leader

Aslak: Secretary

The group leader is responsible for the other members and keeping track of the progress of the group. The same member is also the Scrum master, this means he is responsible for organizing the implementation and development.

The member responsible for external communication is responsible for meetings with project supervisor and product owner. He is also responsible for other types of communication with people outside the group like testers and people we can interview.

Creative/design leader is responsible for the look and feel of the application and will lead the design phase.

The secretary is responsible for taking notes in meetings and discussions. He is also responsible for most of the paperwork.

7.2 Collaboration tools

7.2.1 Facebook Messenger

The group have communicated using the Facebook Messenger. We did not see a need for something more powerful or more tailored to development because the only thing we needed was a way to arrange meetings and ask simple questions. Messenger is also really easy to use because it is something everybody use daily for other things ([Messenger Features](#)).

7.2.2 TeamSpeak3

When the group split up and worked on different things, we sometimes needed to discuss something that was not practical to use messenger for. For this purpose we used

TeamSpeak3, an application letting us talk to each other online. It is originally intended for playing games and letting the members of team-based games communicate, but is also great for other things ([TeamSpeak Systems GmbH](#)).

7.2.3 MeisterTask

MeisterTask is a tool that lets us create backlogs. It works like a digital version of post-it notes and lets us move tasks around in the backlog to signal the status of the task. We used statuses like "In progress", "In testing" and "Finished". Using a tool like this makes it much easier to keep things organized ([MeisterLabs](#)).

7.2.4 GitHub

GitHub is a hub for git version control software. By connecting all the members of the group together in a repository we can easily keep track of the code. It makes it much less of a hassle to combine code from different members and stay updated on the latest version. Because of the version control it is also easy to remove parts of the project if we want to redo something ([GitHub feature](#)).

7.2.5 Google Drive

Google Drive is an online storage platform where you can store any type of file you want. You can also share these files to specific people or make it open to the public([Google Drive](#)). In our project we used Google Docs to keep most of our notes and files for the project, and stored these in a Google Drive. Google Docs is a collaborative writing tool where you can write as a group simultaneously ([Google Docs](#)).

7.2.6 LaTeX and ShareLaTeX

LaTeX is a typesetting language made to write technical and scientific documents ([LaTeX](#)). We used ShareLaTeX as our LaTeX editor. It is an online collaborative editor that allows users to write on one project simultaneously ([ShareLatex](#)).

7.3 Contracts

This project is a serious matter since it is our bachelor project. We decided quite early to make a contract that every member signed([Appendix K](#)). It contained the roles we decided on so that everyone had an area of responsibility and some basic rules. The rules decide the outcome of disputes and what happens if a member does not meet what is expected of him.

7.4 Meeting Times

In our internal contract we also decided on meeting times throughout the process([Appendix K](#)). These meeting times varied from three days a week during the first part of the project to only once a week during development. In reality we probably met more than we ini-

tially planned, especially during some phases of the project we could meet five times every week. Especially during development some of us met at school if they were working on the same thing and wanted to cooperate.

7.5 Logging

Throughout the process we kept two kinds of logs. We logged all our meetings with either product owners or supervisor (Appendix A). We wrote what the meeting was about and followup points on what we should do next. We also logged all our hours worked (Appendix N). Here we also included a small description of what we did for each inputted log. This logging was done separately and put together in the end.

7.6 Discussion

During the planning phase of the project we decided that we wanted to use Scrum as a development methodology. Along with our project plan (Appendix D) we also created a Gantt schema (Appendix L). Part of this Gantt schema is divided into two week long sprints. These sprints covered both the prototyping and development stages of our project.

Our initial plan got changed up a bit due to the idea development phase of the project we had not planned for. During our first planned sprint in the first weeks of February we were focused on research and redefining our idea. We did not have our sprint planning meeting, and was just working on the problems we were up against. This is a trend throughout our process. We worked with an agile mentality, where we tackled the problems we came across and adapted when needed. The biggest example of this is the first weeks on February where we redefined our idea quite a bit because of findings from both research, and discussions.

During the time sprint two should have run we discussed the fact that we did not start our Scrum process properly, and we agreed that we would start the process properly when we started development. In retrospect we see our mistake in this decision. Throughout sprint one and two we did some sketching and we kept up discussions about the system, but it was a month later than we planned where we went 100% on design with Adobe Experience Design. Now we see that this was a point in the process where we had the opportunity to make a choice of putting more focus on either the prototype or development. We could have started development and done some design choices while developing, or just worked on the prototype with more iterations. We were stubborn and wanted to complete the plan we had set for ourselves, which ended up with us not having enough time for either aspect.

Looking back at our design process we see now that instead of discussing as much as we did about the different parts of the prototype, we should have landed on a first iteration earlier and started testing early enough to iterate multiple times. In our prototyping

process we ended up making one iteration that we tested, and brought our insights into the development process. A lot of the problems in the design process are caused by us getting stuck on discussions, in retrospect we see that we should have stopped discussing and let the testers "decide" if it was the correct choice or not. This is where a more set Scrum process where we have a deadline and set time we are going to test would have helped get us going. We should also have Incorporated the end user in the process and gotten more feedback from them other than before the process started. In the end we used as much time as planned on the digital prototyping, but since we started it late, we got a lot less time on our development process.

When we started our development process we had so little time we did not have time to split it up in multiple sprints where we spend time on the meetings in between. So we ended up not using Scrum properly here either, we did have a backlog on MeisterTask, but we did not update it properly, or use it to further our process much. Because we pushed four weeks of development away before Easter, we had to move some time from our report writing period to be able to develop something we could show. Towards the end of the development process we had some overlap where some started on report, while others kept developing. We set a date where we would stop development and only focus on report no matter if we had crucial parts of the solution not yet finished.

8 Ending

8.1 Goals

8.1.1 Development goals

We have to some extent archived our development goals but not to the degree we wanted. Our development goals can be found in section [1.2.1](#). We do have a working prototype for the most important core functionality but not all the functionality we intended. It is also not responsive, but this we did not focus too much on at the moment since we did not have the time in the development phase to implement everything. Responsiveness was among the least prioritized thing on our list since we after the interviews understood that the application would almost only be used in an office on a big screen. We did also not fully use the Scrum methodology to the extent we planned. We did use a wide range of technologies, frameworks and tools. We made the application in 7 different languages, some used more than others. We learned a completely new framework, the Angular framework. We also used a lot of different tools throughout the project, among those we have learned the most from Adobe XD and the use of cli-tools. We have tried our best to follow the industry standard and best practices when it comes to tools and coding, but we have probably diverted a bit since it is not always possible to find the standards, or there are multiple standards. The application is also scalable, it would be easy to eg. add new types of cargo or make new components in angular. We think we did a good job with the UI/UX design, designing with speed and simplicity in mind. We do especially think we solved the layout problems of the application in a good way, keeping it functional and not compromising much on speed and ease of use.

8.1.2 Impact goals

Our impact goals can be found in chapter [1.2.2](#). Our solution as we want it when the prototype would be finished would complete all our impact goals. By using the statistics it would be easy to find qualified transporters since a lot of completed assignments would mean that other companies trust them. The routes would also become more efficient when there is more cargo provided, and our application would do exactly that, provide more cargo for the transporters. By providing more cargo it would also let the transporters fill up their trucks more. By putting the same amount of cargo into fewer trucks we would help decreasing the pollution. And finally the business model of the application would save the entrepreneurs and companies money since the transporters would take the assignments to a lower price because earning some money is better than earning no money.

8.1.3 Learning goals

The learning goals are goals for this project set by NTNU that are supposed to guide us in the project and make us reach for new knowledge ([NTNU i Gjøvik](#)). We think we have completed almost all of them in a satisfactory manner, but may lack a bit in the methodology part of it since we did not fully use the scrum methodology because of the time constraints we made for ourselves since we were too ambitious, just setting it up and making it ready to be used. Otherwise we think that we achieved the goals set for us.

8.2 Future Work

The project is at the moment a design and a prototype containing the core functionality. We imagine two different future versions if this project will continue. The first part is continuing on the current prototype, implementing everything and finishing it. The second part is the fleshing out of the project with more ideas, functionality and the full system we imagine our view to be a core part of.

8.2.1 Finishing the prototype

Profile page

We would like for all users in the backload system to have a profile page. Here, they should be able to provide their contact information and customize their settings. This is something that definitely should be a part of the system.

Login

Every user should have the possibility to register and login to the application in their own created user with their company information related to the user editable.

Notifications

We worked a bit on the notification system and have added some symbols in the prototype and the functional prototype to show where the notifications would have been. We did not implement it, but look at it as a core functionality that should be implemented in the current prototype.

Overview

The overview section is implemented in the design, but it is not implemented at all in the functional prototype.

Bidding

The bidding is not implemented at all in the functional prototype, this must be done before the functional prototype is finished.

Design

The functional prototype do not have much design implemented, this must be implemented in the functional prototype to make it look similar to the design prototype.

8.2.2 Updating table results based on plotted route

A key functionality that is missing is the fact that the results in the table should be directly affected by the route that is plotted from the user. This is not something that we have implemented.

Filtering

At the moment there is only part of the filtering implemented. We implemented the filtering by date and not the filtering by type of cargo.

Hiding/show routes

There is currently no functionality for showing and hiding results for only part of the route. This is functionality we have intended for the final prototype and is added in the design.

Showing routes on map

Showing a generated possible route for the different assignments in the table is functionality added in the design but missing in the functional prototype.

Responsiveness

The site is not responsive, we put this almost last on the list because the application would almost only be used in a office on bigger screens, so for a prototype it would not be that important.

8.2.3 Future functionality

Complaints

What we had in mind was a that for each transportation assignment, a transporter or employer could leave a complaint about the other party if they were unsatisfied. We were thinking that complaints should affect a user's reputation negatively. The reputation we talked about in chapter 5.2.4. A user that has received a lot of complaints would receive some sort of punishment. What we were thinking here was to show an icon next to user's name to indicate that the user is not trustworthy. We did not go much further with this idea because we saw that it would be a time-consuming task to figure out the functionality and details around such a system. It must also be ethically possible to implement in a good way.

Contracts and payment

We did briefly look into the possibility of allowing an employer and a transporter to create and sign a contract directly in Backload instead of having to do this internally between themselves. We were also thinking that payment from the employer to the transporter could happen directly via Backload. The main reason for implementing such a functionality would be to make things easier for the users by not having to use multiple systems. If Backload were capable of handling everything from finding a transportation assignment, to creating and signing contracts and then finally payment, there would be no need to use other systems. However, we decided to not pursue this idea. Just like the idea for complaints, we thought this would be too time-consuming. In addition to this we also saw a big security risk due to our lacking skill set in the area of creating systems that use electronic payment.

Notes

We discussed the possibility of adding a note to a transportation assignment. The main functionality of this note would be to store information the system itself does not store. For example you could write which truck you planned should take the transportation assignment on which route. This would be helpful for remembering if you have are keeping track of a lot of transportation assignments. It would also help communicate this information to your colleagues who might have to take over a bidding war, and needs the original planners intentions with the transportation assignment.

Payment for application

There must be a way for the owner of this system to earn money since it is not free to develop and maintain a system like this. There is multiple possible solutions to earn money on this application.

Adds is the classical approach, but that would probably be a bad solution since it would worsen the quality of the application by taking up space and make it look unprofessional. Some sort of direct payment would make a lot more sense. A one time payment would probably be a bad solution since after a low payment it would be free to use forever. A big payment would probably not be a good idea either, since that would deter the users from buying it. In our opinion, some sort of subscription with a monthly payment, would be much better. This would however not be fair, since the big companies and the small companies would pay the same. We think that the best solution would be a small payment for every assignment, maybe 1% of the final bid. This would make the pricing fair. Those that use the platform the most would then pay the most, and you would only pay for what you actually earn money from.

8.3 Conclusion

Throughout our project we have found answers to many of our research questions. We got our answers through research and working on our design. Some of the questions lost some of their relevance because of our redefinition of scope, like the question about safety regulations.

At the start of the process we set some goals for our project. Even though our project met its fair share of obstacles, we still think we met most of the goals we set from the start. The main obstacle that held us back from meeting our goals was us being too ambitious when defining our scope.

Taking into consideration our ambitions at the start of the project, we are still happy about what we achieved. In retrospect due to the size of the project, we all agree that we should have made a choice before the design process started to either solely focus on design and UX or implementation. This would result in us having the necessary time to focus on creating a more complete solution for a specific part in the application.

Bibliography

About Comments, Feedback, & Ratings. Available at:

<https://www.amazon.com/gp/help/customer/display.html?nodeId=537806>

(Accessed: 26.04.18).

Adobe XD CC. Available at: <https://www.adobe.com/no/products/xd.html>.

Agm-Direction. Available at: <https://github.com/exploosion/Agm-Direction> (Accessed: 15.05.18).

Angular. Available at: <https://angular.io/> (Accessed: 07.05.18).

Angular 2 material implementing sort, filter and pagination. Available at:

<https://stackoverflow.com/questions/45327703/>

[angular-2-material-implementing-sort-filter-and-pagination](#) (Accessed: 14.05.18).

Angular Architecture. Available at: <https://angular.io/guide/architecture> (Accessed: 07.05.18).

Angular CLI. Available at: <https://cli.angular.io/> (Accessed: 07.05.18).

Angular Google Maps (AGM). Available at: <https://angular-maps.com/> (Accessed: 15.05.18).

Angular Material. Available at: <https://material.angular.io/> (Accessed: 07.05.18).

Angular Material Components. Available at:

<https://material.angular.io/components/categories> (Accessed: 07.05.18).

Angular Material Table. Available at:

<https://material.angular.io/components/table/overview> (Accessed: 14.05.18).

Angular Split. Available at: <https://bertrandg.github.io/angular-split/#/> (Accessed: 07.05.18).

Best practices for designing cards – UX Planet (2017). Available at:

<https://uxplanet.org/best-practices-for-designing-cards-a19f53cab052> (Accessed:

11.05.18).

Bing Maps. Available at: <https://www.bing.com/maps> (Accessed: 07.05.18).

Brown, E. (2013). Google maps api v3 - closest places to route within boundary.

Available at: <https://stackoverflow.com/a/17507717> (Accessed: 16.05.18).

- Cao, J. (2017). Web design trends 2016: How cards dominate design. Available at: <https://designmodo.com/web-design-cards/> (Accessed: 11.05.18).
- Cockburn, A. and Highsmith, J. (2001) Agile software development, the people factor, *Computer*, 34(11), p. 131–133. doi: 10.1109/2.963450.
- Cooper, A., Cronin, D., Noessel, C., and Reimann, R. (2014) *About face: the essentials of interaction design*. Indianapolis, IN: John Wiley and Sons.
- Detailed seller ratings*. Available at: <https://pages.ebay.in/help/feedback/detailed-seller-ratings.html> (Accessed: 27.04.18).
- Feedback scores, stars, and your reputation*. Available at: <https://pages.ebay.co.uk/help/feedback/scores-reputation.html#stars> (Accessed: 27.04.18).
- Frances, R., Coughlan, M., and Cronin, P. (2009) Interviewing in qualitative research, *International Journal of Therapy and Rehabilitation*, 16, p. 309–314.
- Geolocation API Specification*. Available at: <http://w3c.github.io/geolocation-api/> (Accessed: 16.05.18).
- GitHub feature*. Available at: <https://github.com/features> (Accessed: 07.05.18).
- Google Docs*. Available at: https://www.google.com/intl/no_no/docs/about/ (Accessed: 14.05.18).
- Google Drive*. Available at: <https://www.google.com/drive/> (Accessed: 14.05.18).
- Google Maps*. Available at: <https://www.google.com/maps> (Accessed: 07.05.18).
- Google Maps Platform*. Available at: <https://cloud.google.com/maps-platform/> (Accessed: 15.05.18).
- How To Design Notifications For Better UX* (2017). Available at: <https://uxplanet.org/how-to-design-notifications-for-better-ux-6fb0711be54d> (Accessed: 28.04.18).
- HTML and CSS*. Available at: <https://www.w3.org/standards/webdesign/htmlcss> (Accessed: 07.05.18).
- Hypertext Preprocessor*. Available at: <http://php.net/> (Accessed: 07.05.18).
- Johnson, C. (2017). Goodbye stars, hello thumbs. Available at: <https://media.netflix.com/en/company-blog/goodbye-stars-hello-thumbs> (Accessed: 26.04.18).

- LaTeX*. Available at: <https://www.latex-project.org/> (Accessed: 12.05.18).
- MeisterLabs. Task management for teams - meistertask. Available at: <https://www.meistertask.com/> (Accessed: 07.05.18).
- Merholz, P. and Norman, D. (2007). Peter in conversation with don norman about ux & innovation. Available at: <http://adaptivepath.org/ideas/e000862/> (Accessed: 07.05.18).
- Messenger Features*. Available at: <https://www.messenger.com/features> (Accessed: 07.05.18).
- MOBGEN (2017). Low-fi prototyping: What, why and how? Available at: <https://mobgen.com/low-fi-prototyping/> (Accessed: 08.05.18).
- Node.js*. Available at: <https://nodejs.org/en/> (Accessed: 07.05.18).
- Norman, D. and Nielsen, J. The definition of user experience (ux). Available at: <https://www.nngroup.com/articles/definition-user-experience/> (Accessed: 07.05.18).
- NPM*. Available at: <https://docs.npmjs.com/getting-started/what-is-npm> (Accessed: 07.05.18).
- NTNU i Gjøvik. Bwu3900 - bacheloroppgave - webutvikling. Available at: <https://www.ntnu.no/studier/emner/BWU3900/2017#tab=omEmnet> (Accessed: 15.05.18).
- Ponto, J. (2015) Understanding and evaluating survey research, *Journal of the Advanced Practitioner in Oncology*, 6(2), p. 168–171.
- Rettig, M. (1994) Prototyping for tiny fingers, *Communications of the ACM*, 37(4), p. 21–27. doi: 10.1145/175276.175288.
- Sass*. Available at: https://sass-lang.com/documentation/file.SASS_REFERENCE.html (Accessed: 07.05.18).
- Schenker, M. (2016). Designing the perfect notification ux. Available at: <https://www.webdesignerdepot.com/2016/07/designing-the-perfect-notification-ux/> (Accessed: 28.04.18).
- Schmidt, C. (2016) *Agile software development teams: the impact of agile development on team performance*. Cham: Springer.
- ShareLatex*. Available at: <https://www.sharelatex.com/> (Accessed: 12.05.18).
- Stack Overflow. Where developers learn, share, & build careers. Available at: <https://stackoverflow.com/> (Accessed: 14.05.18).

Statens vegvesen (2018). Kjøre- og hviletid. Available at:

<https://www.vegvesen.no/kjoretøy/yrkestransport/kjøre-og-hviletid> (Accessed: 15.05.18).

Swanson, R. A. and Elwood, H. I. F. (2005) *Research in Organizations: foundations and methods of inquiry*. California: BK.

TeamSpeak Systems GmbH. Teamspeak 3. Available at:

<https://www.teamspeak.com/en/teamspeak3.html> (Accessed: 07.05.18).

TypeScript. Available at: <https://www.typescriptlang.org/> (Accessed: 07.05.18).

Waze. Available at: <https://www.waze.com/> (Accessed: 07.05.18).

Wilson, C. (2013) *Brainstorming and beyond: a user-centered design method*. Amsterdam: Elsevier.

Økt transportarbeid. Available at: <https://www.ssb.no/transport-og-reiseliv/artikler-og-publikasjoner/okt-transportarbeid>

(Accessed: 07.05.18).

A Meeting logs

11.01.2018

Duration: 1 hour

Present: Group, product owners, project supervisor

Aslak, Christoffer, Mats, Håkon, Johannes, Rolf, Kjell Are

First meeting with between us and supervisor, and product owners and supervisor.

Discussion:

The product owner pitched the idea to supervisor, and supervisor aired some thoughts and suggestions about the application.

Some discussion about contracts and rights.

Follow-up points:

- Research competitors

11.01.2018

Duration: 1 hour

Present: Group, project supervisor

Aslak, Christoffer, Mats, Håkon, Kjell Are

First guidance session with project supervisor

Discussion:

How the UI could look. How a map can play a crucial role in it. Simplicity is key Other similar solutions

Follow-up points:

- Research uber, airbnb, mittanbud, waze. plus any other similar solutions/competitors we can find.
- Work on the project plan.

19.01.2018

Duration: 1 hour 30 minutes

Present: Group, project supervisor

Aslak, Christoffer, Mats, Håkon, Kjell Are

For this meeting we had discussed some potential problems and changes we suggest for the project. We also presented some competitors in the market.

Discussion:

Competitors and how we should proceed with these in mind

- We can either improve upon an existing solution
- Develop a niche product that covers

Problems we found with the solution

- Drivers plan far ahead, and often will not have time to take extra load while out driving.
- Backload then needs to be part of the planning phase instead.

Ethical problems with too many gadgets in the cabin. Restrictions for use when device is moving.

New way of doing our project plan, we need to deliver four files. Gantt skjema, project description, internal contract(group rules etc), and contract with product owner.

We need to figure out exactly what the project entails. What our limitations are, which limitations we need to set for ourselves. We want to have an achievable project that has a result the product owner can put in use. His next step would be to get funding, so a demo version or proof of concept is what we should strive for.

Follow-up points:

- Send email with thoughts, and suggestion to product owner.
- Work on the plan, now that we have a better idea of what is expected.

24.01.2018**Duration:** 2 hours**Present:** Group, product owners, project supervisor

Aslak, Mats, Håkon, Johannes, Kjell Are

Before this meeting we sent a email to product owner with some information about competitors, and potential problems we can see with the solution. The purpose of this meeting is to find the path the product owner wants to go down and figure out how far down that path we should go in this project.

Discussion:

The first part of the meeting was with both product owner and supervisor. We discussed the path of the project. The product owner was very open for us to do what we wanted to work with to get a good grade.

- Bidding rounds
- Open for expansion later.
- Time, rating are also aspects that can affect the choice for the entrepreneur
- Licence plate recognizer, might only be available in some countries

Second part of the meeting was with only the supervisor where we discussed the first part, and what we should do moving forward.

- Finding out the path we want to take in the project.
 - Focus on UX and end up with hifi prototype
 - Focus less on UX and more development and end up with a demo/proof of concept.
 - Which part of the application to focus on. The driver screen, the planners screen, the entrepreneur screen.
- Is there an application that focuses on fleetcontrol. Controlling multiple units, like postal service, or something along those lines.
- Which roles are in our system, and what do these roles want to achieve

Follow-up points:

- Figure out what we want to do in the project
- Work on contracts, both internal and external

08.02.2018

Duration: 40 minutes

Present: Group, product owners, project supervisor

Aslak, Christoffer, Mats, Håkon, Johannes, Rolf, Kjell Are

We had prepared paper sketches of our initial thoughts to show both product owners and supervisor.

Discussion:

We presented our sketches, which we had not put enough time into. We are in the process of redefining the idea.

Follow-up points:

- Have a digital prototype for next meeting.

27.02.2018

Duration: 2 hours

Present: Group, project supervisor
Aslak, Christoffer, Håkon, Kjell Are

Discussion:

We need to find out how external factors impact our solution. Suggestions for what these factors could be:

- Laws and regulations
- Assignments
- Time
- Real-time data
- Solutions already in use in the firms?
- Driver
- Users
- UI conventions
- Existing UI's that have the same features

Building the foundation again, because of many changes and iterations previously in the project.

- Why does x need the system?
- What kind of data does that system take in?
- What should the system do?
- Who is the system for?

Research prototyping tools

- Adobe XD?

Follow-up points:

- For the next guidance we should have written about all of the points above, and researched different prototyping tools

22.03.2018

Duration: 1 hour

Present: Group, product owners

Christoffer, Mats, Håkon, Johannes, Rolf

Discussion:

We showed the prototype to Johannes and Rolf. They were positive.

Follow-up points:

- Johannes thinks it would be a good idea to increase the radius when the drivers are supposed to take brakes, so that the truck can be loaded when the driver is taking his break.

23.03.2018

Duration: 45 minutes

Present: Group, project supervisor
Christoffer, Mats, Håkon, Kjell Are

In this meeting we presented a prototype of the application made in Adobe XD. We discussed some changes that might be necessary to make, and what steps to take further.

Discussion:

We received feedback from our supervisor that the current solution with the map functionality is not good enough. We discussed how we could improve this to better suit routes that are plotted vertically on the map. The current solution works well for horizontal routes, but steps need to be made to make sure that a good experience can be achieved for vertical routes as well.

Suggestions:

- Realign the different components on the “oppdrag” page. This might allow for a more quadratic map, which is a more ideal shape than the more rectangular shape it currently has.
- Make it possible to drag and drop components over on another screen. This would allow for a full screen map on one screen, and a full screen table with data on another.
- Implement drag to resize ratio between map and table plus add full-window view for map

17.04.2018

Duration: 1 hour

Present: Group, project supervisor

Aslak, Christoffer, Mats, Håkon, Kjell Are

In this meeting we presented what we had developed so far, and discussed how we should proceed going in to the last four weeks of the project.

Discussion:

Tips for report:

- Include the whole process in the report.
- Split up process and product for analysis etc.
- Double check the formalities for report structure. IMRaD, frontpage/infopage

Follow-up points:

- Focus only on report from now on

24.04.2018

Duration: 1 hour

Present: Group, project supervisor

Aslak, Christoffer, Mats, Håkon, Kjell Are

We had made a disposition for our thesis and wanted feedback and thoughts on it

Discussion:

Seperating product and process

- Split main bulk of document into two parts with roman numerals
- Seperate process into a chapter of its own at the start or end

Background and idea development chapters

- Kjell suggest background could only be Johannes and Rolf's background and the initial idea presented by them
- Then all research we did into the background and basics of the transport industry comes in the idea development chapter

Follow-up points:

- Get more detailed sections and subsections

09.05.2018

Duration: 1 hour

Present: Group, project supervisor

Aslak, Christoffer, Mats, Håkon, Kjell Are

We had some questions about the report, and wanted some guidance on some of the formal things, and around some general questions about what we should include/not include

Discussion:

We can have our discussion on design/implementation/process chapters, instead of a separate discussion chapter. Its all up to us, but it is not unheard of to not have a discussion chapter.

- Why have we not included the end user more in the design process
- Where we have gotten the code we did not make.
- We have done a project that could/should have been done by maybe two BIXD and two BWU students

Follow-up points:

- Good luck
- Involve Kjell in presentation prep.

B Persona



Leif Are Hansen

- Age: 35
- Man, married, 3 children
- Residence: Raufoss
- Occupation: Planning and logistics at Gjøvik Transport AS

Leif Are enjoys reading crime novels, playing tennis at the local tennis club and spending time with his kids and wife.

He sees himself as a positive and organized person. He enjoys solving the riddle that is controlling a fleet of trucks and trailers.

Gjøvik Transport is a relatively small transportation company with 7 trucks and appropriate equipment to fit them.

Scenarios

Leif Are has three routes in the coming week where two will drive back empty, he uses Backload to find transportation assignments for these.

Leif Are just got to work and there is a note on his desk saying that he needs to keep an eye on three bidding wars on Backload ending today.



Silje Johansen

- Age: 29
- Man, engaged
- Residence: Nydal
- Occupation: Planning and logistics at Hamar Transport AS

Likes to go out with friends, watching TV shows with her fiancé.

She is a positive influence on her colleagues and the workplace in general. She strives to keep everyone motivated.

Silje always says her desk is not messy, but has a system only she knows.

Hamar Transport AS is a large company with over 70 trucks in their fleet.

Scenario

Silje has fifteen trucks with partially empty cargo areas on parts of their route. She wants to fill these up using Backload.



Bernard Jensen

- Age: 32
- Man, single
- Residence: Gjøvik
- Occupation: Owner of Jensens Transport AS

Likes to visit new places and see the world. Loves cars and was originally planning on becoming a mechanic, but after getting his license to drive trucks he got a job as a driver. Eventually he started a sole proprietorship, and has been thriving ever since.

Some would say Bernard is a workaholic. He is really invested in his company but he enjoys it so much. He often jokes that he is married, just to his job.

He gets a good amount of requests for transportation assignments.

Scenario

Bernard wants to regularly check his upcoming routes for smaller extra transportation assignments.



Kåre Lorentzen

- Age: 34
- Man, married, 1 child
- Residence: Gjøvik
- Occupation: General Manager of Entreprenør Lorentzen

Likes to go hiking in the mountains, staying at home with his son and wife and just relaxing with some video games. Driven to improve himself and the company, and is always looking for new and innovative ways of achieving this.

Inherited the company from his retired father who started the company himself at a young age.

Scenario

Kåre needs steelbeams to his construction site in four weeks according to the plan.

C Interview and survey questions

Interview - Transportation company

- How big is the company you are working in? (approximate amount of trucks/drivers/office employees)
- Do drivers always drive the same trucks?
- When a driver has a transportation assignment to transport something from A to B, is the route the driver takes predefined, or does the driver have room to use his/her own experience to choose the best route?
- What type of device is mostly used for the planning job? (PC, tablet, mobile)
- Can you see what the drivers are doing? (where they are, if they take breaks)
- How do you plan the routes?
 - Do you use a computer system for it?
 - How do you delegate the route to the right driver?
- What works well with the current method of planning, do you have anything you would've liked improved?
- How does a normal work day look for you? (Employee that works in the office)
 - How much of your average day goes to planning/organizing transportation assignments?
 - What is the hardest/most timedemanding part of your job?
- How do you keep track of which trucks and trailers are in use, and what their capabilities are?
- Do you contact customers, or does customers contact you?
- What measures do you take to uphold laws, around rest.

Survey - Transportation companies

- How big is the company you work in? (approximate amount of trucks/drivers/office employees)
- How do you plan your transportation assignments?
- Do you use a computer system?
- How do you delegate the transportation assignment to your drivers
- What works well with the way you are planning routes today. What do you want improved?
- How much of your average day goes to planning/organizing transportation assignments?
- How do you keep track of which trucks and trailers are in use, and what their capabilities are?

Interview - Truck driver

- How big is the company you are working in?
- Do you always drive the same truck?
- Do you plan your routes (the way you drive the route) yourself?
 - If so, do what degree are you left to your own planning
- Do you ever bend/break the regulations around rest?
 - If so, which ones, and why?
- How much do you think the resting regulations are broken in general
- Is driving in a convoy a problem?
 - In regards to time constraint or the resting regulation
- Do you use a smartphone / tablet in your work
 - If so, to do what?
- Do you think the resting regulation rules works as they are?
 - What is good/bad?
- Can anyone monitor where you are, if you take breaks in realtime?
 - Who and how?

D Project plan

Background

Every day, thousands of trucks and trailers are used for transporting goods across various routes in Norway. More often than not, this kind of transportation is done by driving from a pickup location (A) to a drop-off location (B). The vehicle is then driven back to the starting location where the route is repeated. This means that there is a large amount of time where these vehicles are driving with empty cargo. SSB says that 30% of the time is spent driving without a load. (SSB, 2017)

Problem description

At NTNU in Gjøvik Johannes Nordsveen presented a problem he had found and a potential proposal for how to solve it. He recognized transportation vehicles often drive with cargo-load only one way, and drive back empty after loading it off. Johannes believed this could be solved by creating a web application for taking on additional transportation jobs that fit with the already planned transportation route.

Goals

The group have come up with a set of goals that we intend to achieve. The development goals are set in relation to not just what we want to create, but also how we approach them. The impact goals are set in relation to what we want the product outcome to accomplish in the long run.

Development goals

- Create a working prototype for the core solution that both the product owner and the group can be proud of.
- The application should be a fully responsive web application that can be used on the majority of all browsers.
- Make use of a wide range of technologies, frameworks and tools.
- Follow industry-standard and best practices.
- Follow the Scrum methodology.
- Make use of good UI/UX design to create a user interface that appropriately reflects the needs of the target user, while also being easy to understand and use.
- The application should be scalable. Should be able to add functionality for other target audiences at a later stage.

Impact goals

- Help entrepreneurs find qualified transportation companies.
- Make it easier for businesses to plan more efficient routes.
- Reduce the amount of empty space in transportation vehicles by filling up the cargo space.
- Help reduce pollution by decreasing the amount of trucks needed to move the same amount of cargo.
- Reduced costs for entrepreneurs and higher profits for businesses.

Research questions

- What other similar solutions exist?
- What will the different user roles in our solution be, and what are the needs for each of the roles?
- How can we make the solution as profitable as possible for both entrepreneurs and companies?
- What existing solution(s)/software do transportation companies use to organize and plan routes, and is it possible to integrate this/these in our product?
- How can we make the application simple and easy to use no matter the amount of vehicles and drivers being managed by each worker in a company?
- What safety regulations are drivers legally required to follow, and how can we take these into account in our solution? Are there any additional steps we can take into our solution to make it as safe as possible?

Methods and means

- Research
- Competitor analysis
- Interview
- Survey
- Prototyping
- Scenarios/personas
- Sketching
- Lo-fi/paper prototyping
- User testing
- Scrum

E Interview consent form



Intervjudeltaker samtykke-skjema

Vi er en gruppe på fire studenter ved NTNU i Gjøvik som studerer Webutvikling og vi holder nå på med bacheloroppgaven vår. Bacheloroppgaven er relatert til transportbransjen, og vi ønsker derfor å samle inn informasjon fra denne bransjen. Opptak av lyd vil ikke bli brukt til noe annet enn dette prosjektet. Det er helt frivillig å trekke seg senere, da vil all informasjon du har avgitt bli destruert. Om det skulle dukke opp noen spørsmål senere kan vi kontaktes ved bruk av kontaktinformasjonen nederst på skjemaet. Takk for at du deltar!

Jeg, _____, som er over 18 år, vil frivillig delta i dette intervjuet, og jeg forstår at informasjonen som blir innhentet under dette intervjuet vil bli brukt videre i bachelorprosjektet.

Jeg tillater at det kan tas lydopptak av intervjuet (lydopptak vil kun bli brukt for å transkribere intervjuet senere, og vil aldri publiseres eller brukes i noen annen sammenheng):

Ja

Nei

Jeg tillater at intervjuet i sin helhet kan publiseres i bachelorrapporten:

Ja, transkriberingen av intervjuet i sin helhet kan publiseres i bachelorrapporten.

Nei, sensitive detaljer og informasjon som kan identifisere meg og/eller selskapet må utelates fra bachelorrapporten.

Signatur: _____ Dato: _____

Kontaktinformasjon ansvarlig for studentintervju:

Navn: Aslak Ihle
Adresse: Granåsbakken 3, 2827 Hunndalen
Telefon: +47 984 65 836
Epost: aslak.ihle@gmail.com

F Redefining scope

External factors

- Laws
 - There are laws for driving and rest times that the drivers can't break, and the application should not encourage breaking these rules. The application should limit the possibilities of breaking the laws.
- Assignments
 - The (jobs) assignments in the app can be quite different. Masses like sand and gravel are shapeless, so it is impossible to measure height and length like normal cargo and it must then be possible to enter weight and volume. Normal cargo have other requirements, and the most important aspects are weight and dimensions. For assignments where there is little weight, the only important aspect is volume. So it must be possible to enter the different aspects of the cargo in an intuitive way.
- Time
 - Assignment can be in the future, so it must be possible for the transport company to not be overrun by useless data. It must be possible to limit the time aspect to the time period the company is interested in.
 - Assignments should be able to planned ahead of time, so one should be able to look at assignments for a longer time period (days, weeks, months). A good solution for navigating and easily changing the time for which assignments should be viewed is important.
- Real-time data
 - We can use real-time data from sources like google maps to provide more information automatically, to help transportation firms make a better choice. Real-time with information about traffic (accidents, big queues) and weather conditions on the road?
- Solutions already in use in the firms?
 - What information the the users get from their current existing systems is important, because we want to compliment this information, not give the same info/useless info.
- Driver
 - Driver needs contact information for eventual delays/problems in the transporting.
- User
 - Transportation companies
 -
 - Entrepreneurs
 - Entrepreneurs might use the application on site and in the office, so needs to be usable from different types of devices (PC/Mobile(?))
 -
- UI conventions
 - multiple call to action
 - contrast
 - reduce form fields

Scope

Who is the system for?

- Entrepreneurs who want something delivered
- Transportation companies who are looking for extra delivery jobs
 - Mostly smaller firms with 10 or less trucks all the way down to sole proprietorships

Why should the different users use the system?

- Entrepreneurs can use the system to get something transported for cheaper than it would usually be (companies are outbidding each other)
- Companies can find jobs that they can add to their existing routes (more optimal driving, decrease the amount of time they are driving with empty cargo).

What are the needs of the different users?

- For entrepreneurs:
 - To have their deliveries come on time
 - Have their goods delivered safe and securely
 - To be able to send a complaint if they believe the transportation has been unsatisfactory or unacceptable
 - Compensation in terms of money or otherwise if the delivery isn't on time or their goods get damaged/lost
 - To get relevant information (i.e. ratings and/or statistics) about the transportation companies that have bid on their delivery request
- For transportation companies:
 - To get relevant information about the goods that need to be transported
 - To be able to communicate to their customer at anytime to inform about unplanned events, for example if the goods can't be delivered on time
 - To be able to mark a delivery as complete and/or fill out what went wrong
 - To be able to appeal a customer complaint if they believe they can't be blamed or if the complaint has false information

What should the system do?

- One should be able to sign up as a user of the system
 - Type of user? (entrepreneur - looking to get something moved, or transport company - looking for jobs)
 - General information/contact information etc
- Login / Logout / Edit user account / Delete user account
- For entrepreneurs:
 - Make it possible to request a delivery
 - Accept a bid for a job assignment
 - Have some form of communication (phone number) option available for the company who has accepted the job

- After a job is completed, the entrepreneur should be able to give a rating to the company who did the job
- For transportation companies
 - View all available job offers
 - Search for job offers that match specific criteria like date, type of cargo, weight and volume
 - Plot a route into a map interface, and see what other job assignments are possible to include in the route?
 - Filter/Sort the job offers by date, type of cargo, weight, volume etc.
 - Make a bid for a job offer
 - View all job offers that the company has won bids for, or are currently making bids for
 - Have some form of communication (phone number) option available for the entrepreneur that posted the job
 - After a completed job, the company (maybe the driver?) should be able to give a rating for the entrepreneur.

What data goes in the system?

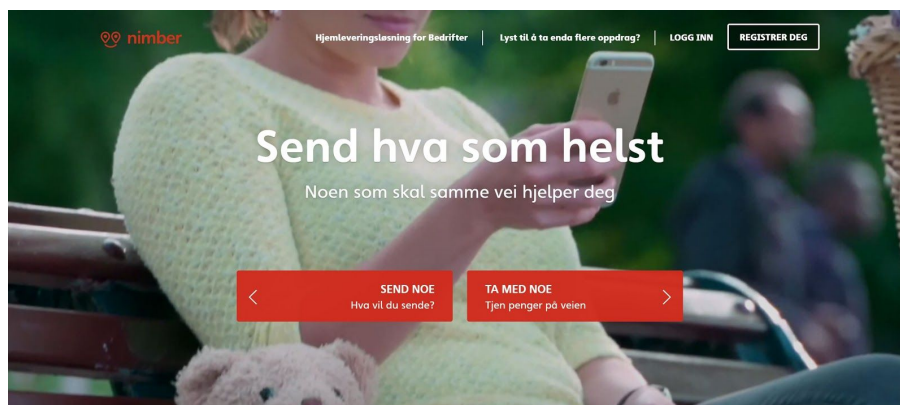
- User data
 - Must sign up (and log in) to use the system
- Transport assignment data
 - Time
 - Type of goods
 - Weight
 - Volume
 - Which entrepreneur gave the specific assignment
 - Rating for that entrepreneur
- Input data (search, filtration, sorting) when transportation companies look for assignments (bedre skrevet)
 - Date
 - Route
 - Time
 - Weight
 - Volume
 - Type of goods
- Data for a bid
 - Rating for company
 - Price of bid
 - Time of when the bid was made?
- Data for a finished delivery job (*vettfaen om det her passe inn*)
 - Completed within time without errors:
 - Time/date of delivery
 - Proof (e.g. image)?
 - Rating for the customer?
 - Incomplete or completed with errors:
 - Explanation of errors

- Reasoning for errors
- Data for delivery complaints (?)
 - Explanation of what was unsatisfactory
 - Proof (e.g. image of damaged goods)
 - Information about the goods cost of repair/replacement(?)
 - Amount of money requested for compensation (?)

G Competitor research

Potensielle konkurrenter og lignende eksisterende løsninger

Nimber.com



Nimber is a solution to transport things between people. It is made for routes one usually drives, typical people driving a distance to and from work that can bring some stuff with them on their daily drives to earn some extra cash.

 **Klassisk gitar**
Oslo → Moss **200 kroner**

Sender **Peter L.**  Må leveres før **18 januar, 2018**

1. Legg inn det du skal sende

Legg enkelt inn ditt oppdrag på Nimber. Vi tar deg steg for steg gjennom prosessen.

 **Christian J.**  **17 januar, 2018 - 200kr** 

2. Motta tilbud om hjelp

Bringere tar kontakt for å fortelle deg når de kan levere ditt oppdrag.

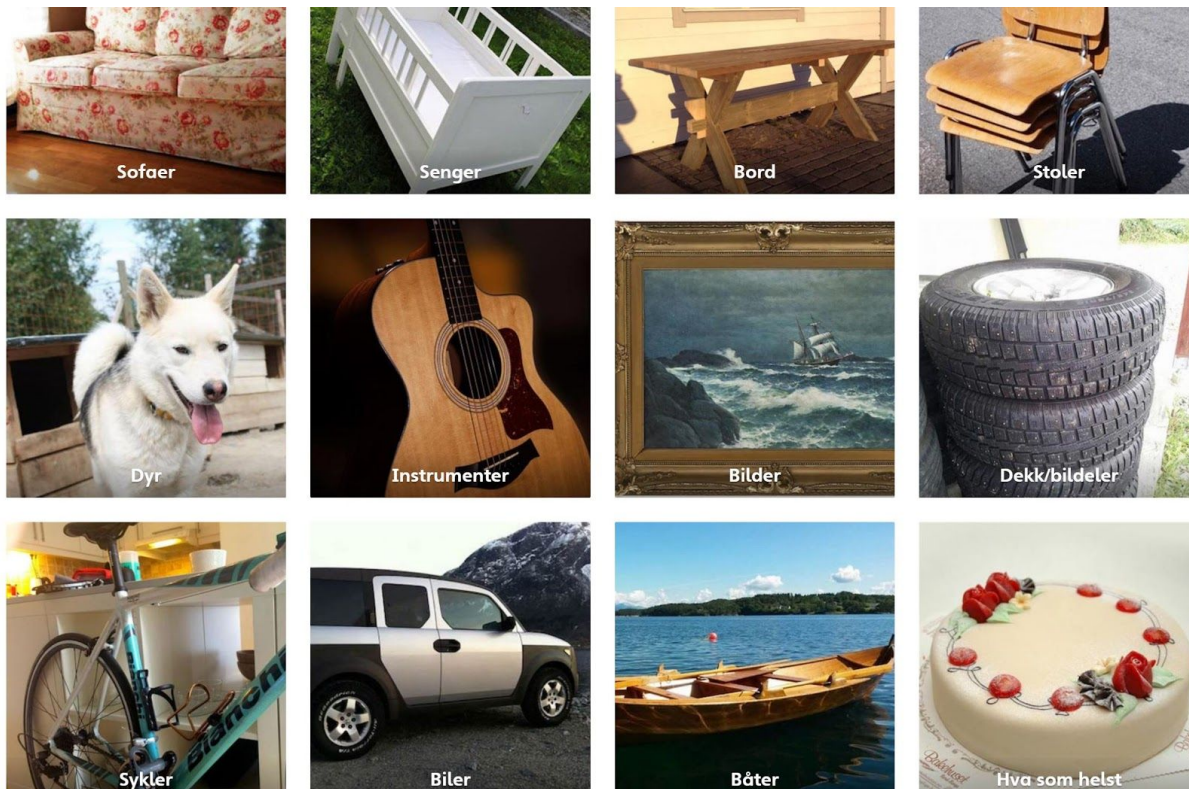
 **Marit K.**  **18 januar, 2018 - 200kr**



3. Gjør avtale om levering

Velg en bringer og bli enig om når det skal hentes og leveres.

You input whatever you want transportet on Nimber with a description and pick-up/drop-off. Then whoever wants to transport it contacts you and you make a deal.



With nimber you can transport most smallsized packages

Uship.com

Uship is a solution where you either send or transport something, it can be almost anything. Anyone can use the solution. Uship is used in the tv-show Shipping Wars.

The screenshot shows the Uship.com website interface. At the top, there are navigation links: [HOW IT WORKS](#), [FIND SHIPMENTS](#), [SHIP](#), [SIGN IN](#), [JOIN](#), and [HELP](#). The main banner is split into two sections:

- I'M A SHIPPER**: I HAVE STUFF I NEED TO GET FROM A TO B.
- I'M A CARRIER**: I HAVE EMPTY SPACE IN MY TRUCK I WANT TO FILL.

Below the banner, there are three categories:

- BUSINESS SHIPPERS**: I WANT TO CONNECT MY BUSINESS WITH USHIP.
- PARTNERSHIPS**: I WANT TO REACH MORE CUSTOMERS AND CARRIERS.
- BROKERS**: I WANT TO REACH MORE CUSTOMERS AND CARRIERS.

A URL is visible at the bottom left: <https://learn.uship.com/shippers/>

You can input what you want transported, and its start and destination, and the info one would need to know if you can take the request (weight, size etc.)

If you want to transport something there is a page where you can see all available listings. On this page you can sort on different categories like location, price etc. You can also see bid information on each listing.

Shipment	Price	Origin	Destination	Kilometers	Ending
7 Items For Transport Furniture Weight: 308.3 kg	14 bids (4 active) Low: \$322 Place Bid	Eugene, OR Residence 10-2 - 12-2	Portland, OR Residence 13-2 - 14-2	177	10h 41m
Totes Appliances Weight: 2721.6 kg	1 bid (0 active) Place Bid	Wasilla, AK Storage Facility 11-1 - 1-4	Vista, CA Residence 31-3 - 1-4	5945	1d 3h
Vibratory Cable Plow Attachment Other Household Goods Weight: 499.9 kg	4 bids (2 active) Low: \$270 Place Bid	Wyoming, DE Business (with loading ... 15-11 - 22-11	Jersey City, NJ Business (with loading ... 22-11 - 22-11	264	2d 4h
Move Drum Sets From Idaho To Sw... Other Household Goods Weight: 90.7 kg	3 bids (0 active) Offer: \$475	Donnelly, ID Business (with loading ... 26-12 - 2-1	Fort Myers, FL Residence 26-12 - 9-1	4583	5d 11h



Question: Hello it's just the attachment? 16-01-2018

Response: Just attachment, strapped on pallet. Thanks 16-01-2018

Response: Ok and it's 6.7ft wide. 16-01-2018

Bids on this Shipment


Price	Service Provider	Service type
\$270	oceansidetrucking no reviews at this time	Blanket Wrap
\$695	Woodsrat (18) ★★★★★ 5.0/5 18 reviews	LTL Standard
\$300 Expired 17-1	PeteW88 (4) ★★★★★ 5.0/5 4 reviews	Blanket-Wrap
\$300 Match Canceled	ty13105 no reviews at this time	LTL-Standard


Members Sign in to Place Bid Join uShip and Place Bid


Truckit.net


TruckIt is an online marketplace for road freight services in Australia. Anyone can ship basically anything as long as it fits in a trailer. If you need to get something moved, you can post it to the website by providing info about the shipment, pickup and dropoff location, time etc.


Support



Fragile Goods



Pilot Drivers



Project / Tenders


Waste Disposal


Hot Shot







2. Enter Item Details:

Is this an auction item or online listing? No

Make and Model:*

Quantity:*

Upload a Picture (max. size 1MB): Ingen fil valgt

Additional Images

[Add another item?](#)

Drivers can search for available freight jobs and filter by category, location and time.

Support

Home
Get Quotes
Search Freight Jobs
How It Works
Blog
More
My Dashboard

Transport Providers

Register for free and start bidding

Register Now!

Location:
Local Search
State Search
Along a Route Search
View All

Find jobs within 10 Km radius of suburb or post code






Collection or Delivery Point
 Collection and Delivery Point
 Collection Point Only
 Delivery Point Only

Keyword Search:

Category:

Select All | Unselect All

- Agriculture Commodities
- Boats
- Cars
- Fragile Goods
- Furniture & Household
- General Freight
- Haulage
- Heavy Equipment

Item	Collection	Delivery	Distance	Quotes	Listed
 Cars hz holden wagon	HAYMARKET NSW 2000 22/01	PERTH WA 6000	3948 Km	0	18 mins ago
 Cars Nissan patrol	WASLEYS SA 5400	CAIRNS QLD 4870	3137 Km	0	24 mins ago
 Cars 2001 Toyota Hilux	ALICE SPRINGS NT 0872	WERRIBEE VIC 3030	2236 Km	0	25 mins ago
 General Freight Side rails and steps for a 4WD car	PORT MACQUARIE NSW 2444	TOLGA QLD 4882	2243 Km	0	27 mins ago
 General Freight combi oven rack for combi oven	WEST RYDE NSW 2114	KINGAROY QLD 4610	1010 Km	0	28 mins ago

Loadshift.com.au

Loadshift is an online marketplace for transport carriers and shippers, carriers can post their current or future truck/trailer location and availability, and shippers can post shipping requests or find suitable trucks on a location map. This allows carriers to choose listed transport requests while also receiving job leads. Carriers have to pay a monthly fee to access the transport requests list, which is how Loadshift earns money.

1300 562 374 | info@loadshift.com.au

Loadshift 208,287 km of requested transport added in the last 24hrs. MEMBERS LOGIN

HOME ABOUT US HOW IT WORKS CARGO CATEGORIES PRICING CONTACT US GET QUOTES GET LOADS FIND TRUCKS

User Filter: All | Pickup State: All | Delivery State: All | Cargo: All | Order by: Newest to Oldest | Refresh Requests

★ = Favourite GREEN = New Job URGENT (ASAP) RTG = Ready To Go (RTG) Next 24/48hrs (24/48hrs) View Details

ID	Pickup Suburb	Pickup State	Delivery Suburb	Delivery State	Cargo	Est/Km
270697	South west rocks	NSW	Green Point	NSW	Houseboat	392 km
270696	Coffs Harbour	NSW	Sandgate	QLD	Hyundai iLoad (not running)	411 km
270695	Streatham	VIC	Goondiwindi	QLD	CAT MT845 Challenger	1,422 km
270694	Wagga Wagga	NSW	Gympie	QLD	9.5mtr Iveco 8wheeler Water Truck	1,353 km
270692	Cairns	QLD	Lytton	QLD	2x hiab type cranes on pallets + small power pacs	1,692 km
270691	Tullamarine	VIC	Loganholme	QLD	CAT 815 Compactor - Wd:3.5mtr	1,736 km
270690	Munruben	QLD	Tamworth	NSW	Nissan Xtrail	565 km
270689	Milnerra	NSW	Wallerah	NSW	Empty 20ft Container	117 km

Screenshot of the transport requests list where carriers find jobs.

1300 562 374 | info@loadshift.com.au

Loadshift 208,287 km of requested transport added in the last 24hrs. MEMBERS LOGIN

HOME ABOUT US HOW IT WORKS CARGO CATEGORIES PRICING CONTACT US GET QUOTES GET LOADS FIND TRUCKS

User Filter: All | Pickup State: All | Delivery State: All | Cargo: All | Order by: Newest to Oldest | Refresh Requests

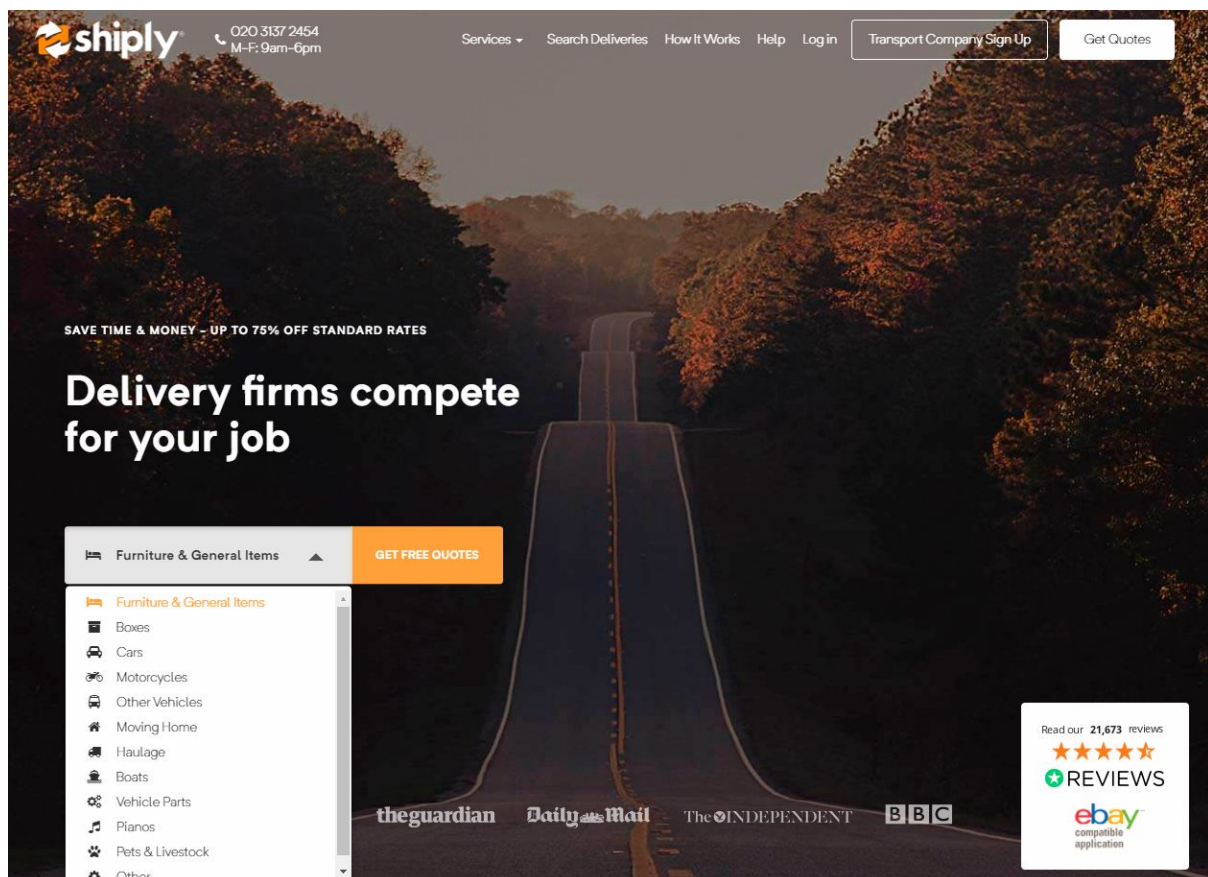
★ = Favourite GREEN = New Job URGENT (ASAP) RTG = Ready To Go (RTG) Next 24/48hrs (24/48hrs) View Details

ID	Suburb	State	Stamp	TruckType	Comments
46368	BOURKE	NSW	16/01/18	8MTR FLAT BED	BEAVERTAIL TRUCK AVAILABLE FROM THE 16.01.18 RETURNING TO BRISBANE/GOLD COAST FROM BOURKE.
46413	Brisbane	QLD	16/01/18	Tri axle deck widening low loader	Have a tri axle deck widening low loader (to 3.5m) available for loading in the Brisbane area on Tuesday 16/1/18 looking to load to Sydney or [more...]
46417	Carnarvon	WA	16/01/18	Prime Mover/Float	We are dropping off 150 south of Carnarvon tomorrow the 18th January 2018 & am looking for a load heading to Broome
46419	Wetherill Park	NSW	16/01/18	32 pallet b/d	Looking for load to Brisbane 32 pallet tautliners
46410	Karatha	WA	17/01/18	P/Mover & 37Ton Float	Looking for load out of Karratha/Port Hedland, Heading back to Broome Drake 37ton Spreader Float
46415	Brisbane	QLD	17/01/18	Prime mover and drop deck	We have a 43ft drop deck widening trailer with ramps looking to fill our bottom deck, 9.4m long, with a load heading to preferably

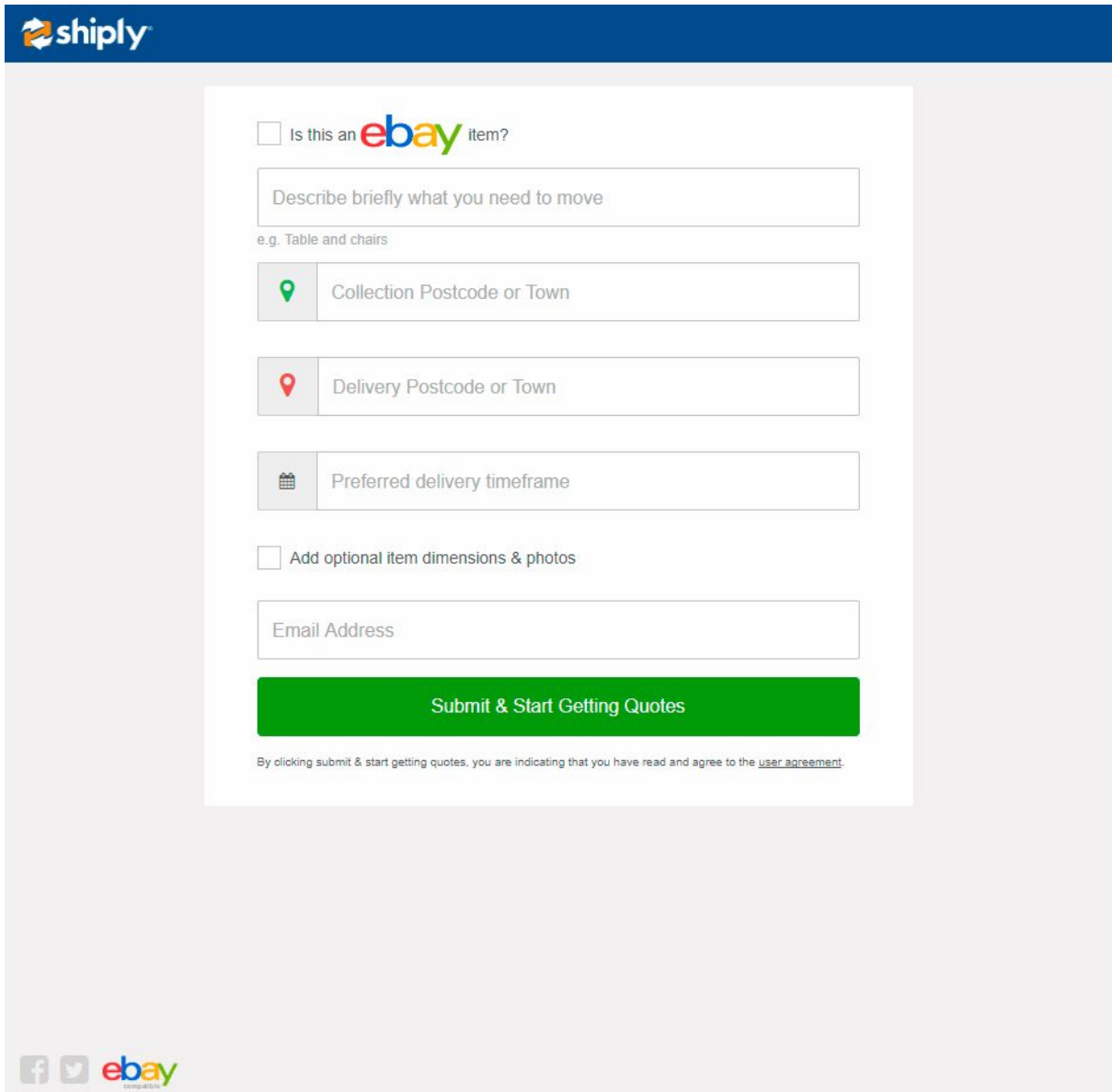
Screenshot of the available trucks/trailers list where shippers can request services.

ShiPLY.com

ShiPLY is an online marketplace for transporting goods. Anyone can list items they need moved, and trusted delivery companies will send price offers for transporting the item. Courier companies use shiPLY as a way to keep their transport vehicles full of cargo along routes they are already traveling. This means courier companies earn more money by utilizing otherwise empty space, and customers pay less for transporting items.



Screenshot of the ShiPLY homepage.



The screenshot shows a web form for Shiplly, a company that partners with eBay. The form is set against a blue header with the Shiplly logo. The form itself is white and contains several input fields and checkboxes. At the top, there is a checkbox labeled 'Is this an ebay item?'. Below this is a text input field for describing the items to be moved, with an example 'e.g. Table and chairs'. This is followed by three more input fields: 'Collection Postcode or Town' (with a location pin icon), 'Delivery Postcode or Town' (with a location pin icon), and 'Preferred delivery timeframe' (with a calendar icon). There is another checkbox for 'Add optional item dimensions & photos'. Below these is an 'Email Address' input field. A prominent green button labeled 'Submit & Start Getting Quotes' is at the bottom of the form. A small disclaimer at the very bottom of the form states: 'By clicking submit & start getting quotes, you are indicating that you have read and agree to the [user agreement](#).' In the bottom left corner of the page, there are social media icons for Facebook and Twitter, and the eBay logo with the text 'comp. able to application'.

Is this an **ebay** item?



Describe briefly what you need to move

e.g. Table and chairs

Add optional item dimensions & photos

Submit & Start Getting Quotes

By clicking submit & start getting quotes, you are indicating that you have read and agree to the [user agreement](#).

  **ebay**
comp. able to application











Screenshot of the form customers need to fill out for transporting items.

Are you a transport company? You can quote on this work with no sign up or subscription fees.

[Learn More](#)

Results: 1 - 50 of 9,510

1 2 3 > >>

Item	Mileage	Date	Quotes
 20ft X 8ft Empty Office Container Chipping Sodbury, South Gloucestershire, BS37 Chalfont Saint Giles, HP8	106	05/01	10
 Designers Guild Leopold Chartreuse Rug... URGENT London, Greater London, SW15 Elgin, Moray, IV30	584	yesterday	0
 Kenwood Stainless Steel Dishwasher Nottingham, Nottinghamshire, NG7 Cambridge, Cambridgeshire, CB1	96	05/01	29
 2x Nitro Rc Car And Parts Downham Market, Norfolk, PE38 Wythenshawe, Greater Manchester, M22	164	9 secs ago	0
 Cylinder Head Goldthorpe, South Yorkshire, S63 Minehead, Somerset, TA24	244	32 secs ago	0
 1st Floor Flat Moving. We Need A Luton... Hemel Hempstead, Hertfordshire, HP1 St Albans, Hertfordshire, AL3	9	1 min ago	0
 Honda CRM250AR Broughshane, Ballymena, BT43 Reigate, Surrey, RH2	494	2 mins ago	0
 Hemnes Ikea Day Bed - White With 2 Mat... Chelsea, Greater London, SW1W Havant, Hampshire	69	3 mins ago	1
 Skoda Octavia Sandycroft, Flintshire, CH5 Oldham, Greater Manchester, OL4	57	3 mins ago	0
 Habitat Hendricks Blue Fabric Sofa			

Screenshot of the list of transportation requests that courier companies use.

HaulageExchange.co.uk

Haulage Exchange is a way for haulage and courier companies to work together so they can operate more efficiently and earn more profits. If a haulage/courier don't have enough space or time for a delivery, they can outsource (some of) it to other companies. Which means if a haulage/courier have extra space or time along a route, they can take on extra cargo.

News [Why we should not fear AI and how it can change the freight ...](#) +44 (0)20 8993 7100 [Login](#) [Register](#)

HAULAGE Exchange Solutions [About](#) [Available Loads](#) [Pricing](#) [Case Studies](#) [Contact us](#)

Find loads faster, sell your capacity more easily...

<p>Active users 17,112</p>	<p>Feedback items posted last month 52,897 from members</p>	<p>Load response time 4 min. avg. this month</p>	<p>Avg. load miles 10.2 million per month</p>
--------------------------------	---	--	---

smarter visibility safer growth faster results member miles

Haulage Exchange, for a smarter haulage and logistics business

[Request a Callback](#)

Who are you?

- A Haulage Company
- An Owner Operator

What we have for you

[Available Loads - UK](#) [Available Loads - Euro](#) [Exchange Statistics](#)

Screenshot of the homepage.

News Why we should not fear AI and how it can change the freight ... +44 (0)20 8993 7100 Login Register

HAULAGE Exchange

Solutions About Available Loads Pricing Case Studies Contact us

What are you looking for?

Available Loads

Here's our current list of live, real-time loads available on our Exchange – the quick, one step route to more work, more vehicles, more opportunities.

The Exchange will help you build important connections and contracts and eliminate dead mileage, allowing you to grow your business without incurring further overheads. Available loads for delivery work are placed on the Exchange in real-time. In an average month 100,000 loads are posted on the Exchange and thousands more traded amongst our members directly. What are you waiting for get trading now!

UK	Euro		
From: London E4 UK To: Cannock WS11 UK	Pickup: 19.10 18/01 Deliver: 23.05 18/01	13.6M	View Details
From: Westburn Dr G72 To: Ryton NE40 3... UK	Pickup: 07.30 19/01 Deliver: 13.05 19/01	7.5T	View Details
From: Bridgend CF3... UK To: Feltham TW6	Pickup: 12.00 19/01 Deliver: 15.20 19/01	18T	View Details
From: Pontyclun CF... UK To: Deeside CH5 UK	Pickup: 06.00 19/01 Deliver: 11.25 19/01	7.5T	View Details
From: Alfreton DE5... UK To: Ipswich IP1 UK	Pickup: 08.00 22/01 Deliver: 13.40 22/01	18T	View Details
From: Leicester LE... UK To: Milton Keyne... UK	Pickup: 08.00 19/01 Deliver: 13.45 19/01	7.5T	View Details
From: Brierley Hil... UK To: Glenrothes K... UK	Pickup: 20.30 18/01 Deliver: 03.35 19/01	7.5T	View Details

Screenshot of the “exchange” list.

Mulige problemer og løsninger

1.

Etter en samtale med en sjåfører har vi funnet ut at sjåfører planlegger langt frem i tid, ofte flere uker, og sjelden kommer til å ha tid til å plukke opp last “on the fly” med Backload.

De har begrenset tid fordi:

- Lastebilen skal tilbake til hovedkvarteret for å bytte sjåfører eller starte ny rute
- Neste oppdrag starter til en bestemt tid
- Man har en plan man skal følge for å rekke alt
- Vedlikehold av bilen er planlagt
- Arbeidsdagen er allerede planlagt så effektivt at man ikke har tid til å plukke opp last.

2.

Lastebilsjåføren er ansatt i en bedrift og kommer ikke til å tjene mer ved å benytte backload siden sjåføren som regel har fast timelønn eller fast lønn for hver rute (f.eks Bergen-Oslo). Det vil være bedriften som tjener mer ved å benytte Backload. På grunn av dette er den

parten som er mest interessert i å bruke Backload bedriften. Et kompromiss kan være høyere lønn til sjåførene siden selve bedriften vil tjene mer ved å kjøre mer effektivt. Unntaket er de som driver selvstendig, da er bedrift og sjåfør samme person.

3.

På grunn av sikkerhet bør man unngå at sjåføren skal navigere i en app under kjøreturen siden det tar fokus vekk fra veien og kan skape farlige situasjoner. Sjåføren vi har snakket med har gitt uttrykk for at sjåfører ikke liker å stoppe for å gjøre ting f.eks på digitale skjermer, og at dette ofte blir gjort i fart. De siste årene har det blitt mer vanlig med trådløst headset for telefon, dette er et sikkerhetstiltak for å unngå telefonbruk. Man bør så langt det lar seg gjøre unngå at Backload brukes i lastebilen til formål der man må trykke på skjermen.

<https://www.nrk.no/sognogfjordane/vil-ha-vekk-pc-ar-og-vimplar-fra-frontruta-1.13862782>

<https://www.nrk.no/viten/bilteknologi-kan-fore-til-ulykker-1.11273295>

Våre løsningsforslag:

Forslag 1:

Backload bør brukes av de som planlegger rutene til sjåføren og arbeides inn som en del av planleggingen. Da blir det 3 parter i systemet og backload komplementerer megleren:

1. Entreprenører legger ut det som skal fraktes i backload
2. Selskapet/manager (den som planlegger) tar imot passende tilbud fra backload basert på rutene som allerede kjøres, og utarbeider rutene i henhold til endringene.
3. Sjåfør får kart, rute og informasjon på tablet/telefon.

Forslag 2:

Man kan lage en nisje (f.eks de som transporterer jord, sand og grus), og brukergruppen blir da enmannsforetak som gjerne planlegger i kabinen og har full kontroll på sin egen plan og da kan ta oppdrag "on the fly".

Vår mening:

Vi tror forslag 1 vil være den beste løsningen. Med forslag 1 når man ut til en større brukergruppe, og man utestenger heller ikke enmannsforetak siden de da vil ha rollen som både manager og sjåfør. Det blir da også større miljø-besparelser og profitt siden brukergruppen blir større. Med forslag 2 begrenser man brukergruppen og vil da nå ut til færre sjåfører.

Hva tenker du om forslagene vi la frem?

Skal Backload være en konkurrent til eksisterende løsninger eller være en nisjeløsning?

Backload kan gå begge veier siden det ikke er et selskap som (etter hva vi har funnet ut) er sterkt etablert i Norge, og fordi at backload fokuserer på tungtransport.

H Usertesting consent form Samtykkeskjema

Dette er et samtykkeskjema for brukertesting av designet en bachelorgruppe på NTNU Gjøvik lager til sin bacheloroppgave. Gruppen består av Håkon Horvei, Aslak Ihle, Mats Åsbu Jacobsen og Christoffer Seljehaug.

Vi kommer til å observere og notere hva du foretar deg for at vi i bachelorgruppen skal kunne forbedre designet vårt.

Jeg forstår at de som holder testen kommer til å observere meg og notere hva jeg foretar meg.

Jeg gir bachelorgruppen tillatelse til å bruke observasjonene og notatene i internt arbeid med det formål å forbedre designet som blir testet.

Signatur: _____

Dato: _____

Takk for at du deltar i brukertesting vår.

I Prototype testing tasks

Scenario: Du jobber for transportfirmaet "Bråten Grus og Sand AS" og står for organisering og planlegging av transportoppdrag. Bedriften din har en rekke biler som kjører ulike ruter, men mange av disse bilene har muligheten til å plukke opp ekstra lass. Du skal bruke Backload for å fylle opp lastebilene som har ledig plass.

Task 1: En av bilene skal transporte jernbaneskinner fra Gjøvik til Lillehammer neste uke, lastebilen har mye ledig plass. Bruk Backload for å finne et passende oppdrag til ruten som skal kjøres og legg inn et bud.

Task 2: Du vil nå se etter flere ting du kan ta med deg på ruten, men du vil nå se hele listen og er ikke lenger interessert i å se kartet. Hvordan får du sett den store listen?

Task 3: Neste dag får "Bråten Grus og Sand AS" et nytt oppdrag utenfor Backload, som kan passe bedre med oppdraget du bydde på i går. Legg til et nytt bud for en ny dato, men ikke fjern det første budet du ga.

Task 4: Det har gått en uke og du har fått varsel om at du fikk oppdraget du bydde på. Du har noen spørsmål angående transporteringen, finn kontakinfo til oppdragsgiveren.

Task 5: Du vet at lastebilen er full den første delen av ruten og er ikke interessert i oppdrag langs den delen. Kan du fjerne oppdrag langs den første delen av ruten?

Task 6: Du er på "Finn oppdrag"-siden og har funnet et oppdrag som kan passe men du er ikke helt sikker på hvor hentestedet og leveringsstedet befinner seg i forhold til ruten du har søkt på. Hvordan ville du ha gått fram for å se dette?

Task 7: Du ser et oppdrag som er interessant for deg, men som du ikke kan binde deg til enda ved å gi bud. Lagre dette oppdraget som interessant.

Task 8: Du finner en lapp fra forrige skiftleder: "Jeg har forresten lagret noen oppdrag du burde se på." Finn oppdragene han snakker om.

J Prototype testing result

Test 1

Task 1:

Task 2:

Task 3:

Task 4:

Vanskelig å finne

Task 5:

Vanskelig å tolke teksten, men skjønte hvor det var og hva det gjorde.

Task 6:

Håpløst å finne

Task 7:

Stjerne!

Task 8:

Fant dem fort.

Test 2

Task 1

Tenkte “minimer kart” var animert scroll. Stusset litt på man må bruke et annet system for å se data for ruter (f.eks vekt osv), mente det burde hentes automatisk. Ville trengt notatblokk for å skrive ned vekt osv. Forstod med en gang at man får notification om man får oppdrag.

Task 2

Obvious

Task 3

Stusset litt over sidebar i “Min oversikt”, tenkte først at aktive oppdrag var gitte bud. Når han fant budsiden så forstod han å gi nytt bud for ny dato med en gang.

Task 4

Fant med en gang kontaktinfo, og forstod oppdrag ville ligge i “aktive oppdrag”. Spurte hvorfor han ville flytte oppdrag til arkiv.

Task 5

Hva skjer om jeg trykker på “skjul delrute”?

Task 6

Tenkte at om han kunne trykke på stedsnavn/gatenavn for å vise på kart.

Task 7

Fant med en gang

Task 8

Fant med en gang. “Lagrede oppdrag er kanskje ikke like intuitivt som f.eks favoritter”. Forstod at knappen/ikonet er en toggle.

Etter tasks

Fixed kart kan være problematisk når man åpner “mer info” for oppdrag i tabell. Ellers så er det å hente automatisk data og integrere i tabell, og “fade ut” oppdrag som ikke er relevant basert på vekt du ikke kan ta f.eks.

Test 3

Task 1

Trykket på “vis på kart” først.

Task 2

Forstod med en gang.

Task 3

Trykket på “min oversikt”, tok litt tid å finne “budrunde” som riktig side. Tenkte oppdrag “utenfor backload” kanskje ikke er så relevant.

Task 4

Fant med en gang(?)

Task 5

Fant med en gang etter vi forklarte at det skulle skje på “finn oppdrag” -siden. Kanskje bedre å sette et punkt (klikke på) kartet for å markere. Annet navn enn “skjul delrute” eller checkbox som f.eks “Har ikke last”. Er ikke så lite forklarende som det er nå.

Task 6

Trykket på “vis mer” under tabellen. Tenkte at det burde være mulig å trykke på adressen. Slet veldig med å finne “vis på kart” ikonet. Kanskje mer logisk å flytte plassering på “vis på kart” ikon og “favoritt” ikon, da er “vis på kart” ikon nærmere adressen når man expander. Kanskje ikonene skal være blå.

Task 7

Fant med en gang.

Task 8

Fant med en gang.

Etter tasks

For liten skrift, spesielt på tabell header. Liten skrift og ikon på “arkiv”. Litt overvelmende mye info når man først ser tabellen, men forståelig når man leser. Kanskje det er bedre med å dragge kart/tabell høyde istedet for knapp som det er nå. Kanskje at man lagrer biler hvor man skriver inn hvor mye plass bilen har, så kan man velge bil å bruke i en rute. Flytte backload til høyre, sånn at det er plassert riktig i forhold til “varsel” og “brukernavn” på høyresiden. Flytte oppdrags tittel/beskrivelse/navn i arkiv fra midten til venstre, eller beholde i midten og ha mindre luft under. På budrunder ha mer lik tekststørrelse. Ha header for alle elementer øverst i “budrunder” cards, f.eks “budfrist” som header og dato under.

Test 4

Task 1

Sleit litt med å finne “legg inn rute”, startet først med å bla gjennom tabellen å finne manuelt. Fant ut hvordan man legger inn bud i expand meny.

Task 2

Fant ut etter at vi spurte “Du har ikke lyst til å se kartet. Hvordan gjør du det?”.

Task 3

Vanskelig å finne ut at man skulle gå på “min oversikt” siden for å legge inn for ny dato. Ville først skrive inn ny dato og sånn for å legge inn nytt søk. Måtte lete seg litt fram på oversikt siden.

Task 4

Lette først litt i tabellen for å se informasjon der. Gikk etter hvert inn på min side for å lete der. Fant det etter litt leting på min side.

Task 5

Vanskelig å tolke teksten, men skjønnte hvor det var og hva det gjorde. Hadde

Task 6

Håpløst å finne

Task 7

Stjerne!

Task 8

Fant dem fort.

Test 5

(fikk trykke litt fritt før tasks)

Task 1

La inn dato og gikk deretter til tabell uten å legge inn rute først. Etter å ha lagt inn rute trykket han på expand først.

Task 2

Forstod med en gang

Task 3

Trykket inn på “min oversikt” med en gang, tok bittelitt å finne “budrunde” siden fordi den ikke var førstesiden (aktiv var førsteside).

Task 4

Fant ganske fort. Kanskje hatt card-boksen mindre og hatt kontaktnfo på høyresiden ved siden av resten av innholdet.

Task 5

Selv om han så på knappen/rute-inputs så fikk han ikke med seg “skjul delrute”, fant ut etterhvert. Mener teksten på knappen er tydelig i hva den betyr. Ble litt skjult på grunn av at den er så grå, en hovereffekt hadde kanskje hjulpet.

Task 6

Trykket på “min oversikt”. Trykket på vis på kart ganske fort etter at han spurte om ruten på kartet er “original ruten”, når det ble bekreftet forstod han at “vis på kart” knappen var løsningen. Mente knappen var tydelig og åpenbar, ingen problemer med at den var for lite synlig. Kanskje gjøre teksten større for “vis på kart”.

Task 7

Fant lagre ikonet med en gang. Syntes stjerne ikonet fungerer bra, “er det jeg er vandt med”.

Task 8

Fant med en gang. Forstod hvordan man fjernet oppdrag med en gang.

Etter tasks

Syntes kart/tabell forholdet fungerer greit, ser for seg at drag kunne vært brukbart om man skulle sett begge views samtidig. Tenkte at man kunne selecte flere på “last”. Knapp for å resette rute-søket. Syntes det hadde vært bra om mappet oppdaterte seg etter man har fylt ut “startpunkt” og så vist rute etter å ha fylt ut “destinasjon” fordi det er feedback som viser at man har skrevet inn riktig (og at “ting skjer”). Forslag til alternativt/nytt layout: Dele kart og tabell i to kolonner og vist kartet til venstre og tabellen på høyre. Fordi da ble kartet mer kvadratisk, men tenkte kanskje tabellen ville bli for liten da.

Om tredelt layout (kart/card/tabell): Tror han heller ville ha likt layoutet slik som det er nå, fordi det er en mer naturlig sammenheng. Hadde vært litt forvirrende/rart med cards som oppdatere ut i fra tabellen som er plassert en annen plass.

Likte viewen som er kart/tabell split, for da kan man mer effektivt se oppdrags-ruten på kartet. Syntes knappen som er nå er bra fordi den er "clean" og minimalistisk, enkel og forstå og bruke. Er ikke så glad i drag egentlig, spesielt om det er noe man må gjøre ofte, blir fort slitsomt da. Tror han hadde brukt split uavhengig av skjermstørrelse, for han ville likt å sett hvor ruten (i tabellen) er på kartet. Ville hatt muligheten på mindre skjermer som iPad også, men tror egentlig ikke han hadde brukt det fordi slike ting fungerer ofte dårlig på iPad.

Test 6

(fikk trykke litt fritt før tasks)

Task 1

Fant dato/rute inputs med en gang. Forvirrende at inputs ikke viser "hva de er" etter å ha fylt info, burde kanskje ha floating input text (som i material design). Lurte på hva tabell-infoen egentlig var, om det var andre som hadde lagt inn eller om det var dine egne ruter.

Task 2

Forstod med en gang.

Task 3

Ville ha lagret for å finne oppdrag noen dager senere. Etter å ha forklart bedre hva vi egentlig spurte så gikk han på "min oversikt" og tenkte at budet han hadde gitt ville ligge i "historikk". Fant "gi nytt bud" relativt fort etter han fant riktig side "budrunder"-siden.

Task 4

Fant med en gang.

Task 5

"Kunne jeg ikke bare ha fjernet punktet fra ruten?" (Istedet for Gjøvik -> Lillehammer -> Hamar så tenkte han å bare fjerne f.eks lillehammer om han hadde last fra gjøvik til lillehammer). Fant "skjul delrute" knappen raskt.

Task 6

Forstod med en gang. Om hva han trodde ville skje når man trykker på "vis på kart": Tenkte at det røde ikonet på kartet (destinasjon ikonet) ville flytte seg til den man trykket på.

Task 7

Forstod med en gang.

Task 8

Forstod med en gang.

Etter tasks

Syntes "skjul delrute" var litt clunky. Syntes også at det burde være lettere å legge inn flere bud med en gang (i tabellen), det burde være mulig å legge inn i tabellen sammen med første bud i stedet for å måtte gå til "budrunder"-siden for å legge inn flere. Kanskje ha en "retur" knapp for å legge til ruter som skal kjøre til et punkt og tilbake, i stedet for å måtte legge inn samme punkt to ganger for start- og slutt punkt.

Om tabell drag: Kan være brukbart for de som trenger det, men tror ikke det hadde vært krise å ikke ha. Kan se for seg at det er lettere å få oversikt over rute om man kan dragge til større kart.

Om å gjøre responsivt på mobil/mindre skjermer: Tabell kan/må bli liste i stedet for tabell, for det er ikke plass til alle kolonnene. Kart/tabell må være fullheight toggle, er ikke plass til en delt view. Endre fra tabell til noe mer lignende card.

Om tredelt layout: Det kan være bra, da slipper man å scrolle for å se expanded info. Men samtidig så er det veldig mye info å vise i card-containeren, f.eks om man markerer flere i tabellen så vil det jo ikke være plass. Er ikke helt sikker på om det er en bedre løsningen, f.eks det kan være et problem at man ikke får knyttet tabell expand menyen med ekstra-info i card. Liker bedre sånn som det er nå.

K Internal contract

Internal contract for bachelor project

Internal contract between members on the bachelor project at NTNU Gjøvik that have the task to create a prototype for the Backload-project. This contract is valid until 05.06.18.

Roles and responsibility

- Scrum master/group leader: Håkon
- Communication contact person: Christoffer
- Creative/design leader: Mats
- Secretary: Aslak
- Supervisor: Kjell Are
- Product owner: Johannes Nordsveen, Rolf Schonhowd

Project rules

- If a group member cannot attend a planned meeting, inform the other members as soon as possible.
- If a group member regularly do not put in the agreed amount of work that is expected, he could be removed from the group. This would require multiple warnings (three) and must be a unanimous decision between the other group members and the project supervisor.
- Potential costs should be split equally among all group members.
- Documentation and code should follow the set of rules and standards agreed by the group.
- If the group can not agree on a decision in a reasonable time, the group leader have the ability to set the decision on hold till later. If the decision still can't be agreed upon on a later time, it will be decided by voting where the group leader has two votes.

Meeting times

At the start of the project we want to meet regularly, because we need to discuss a lot of different things in the starting phase of the project. Later in the project when we are running sprints and the tasks are more individual we will not have the same amount of regular meetings in person, but have a daily standup meeting (in person or digital) so everyone knows what everyone is working on that day. And we will meet in person at least once a week to get the more general idea of what people are working on that week.

Phase 1 (Planning), Phase 2 (Design), Phase 4 (Report) and Phase 5 (Presentation):

- Mondays: 10:00 - 1400 (+)
- Wednesdays: 10:00 - 15:00 (+)
- Fridays: 10:00 - 13:00 (+)

Phase 3 (Development):

- Weekly meeting: Monday 10:00 - 12:00 (+)
- Daily meeting: Tuesday - Friday 10:00 - 10:15

Signatures

Aslak Ihle

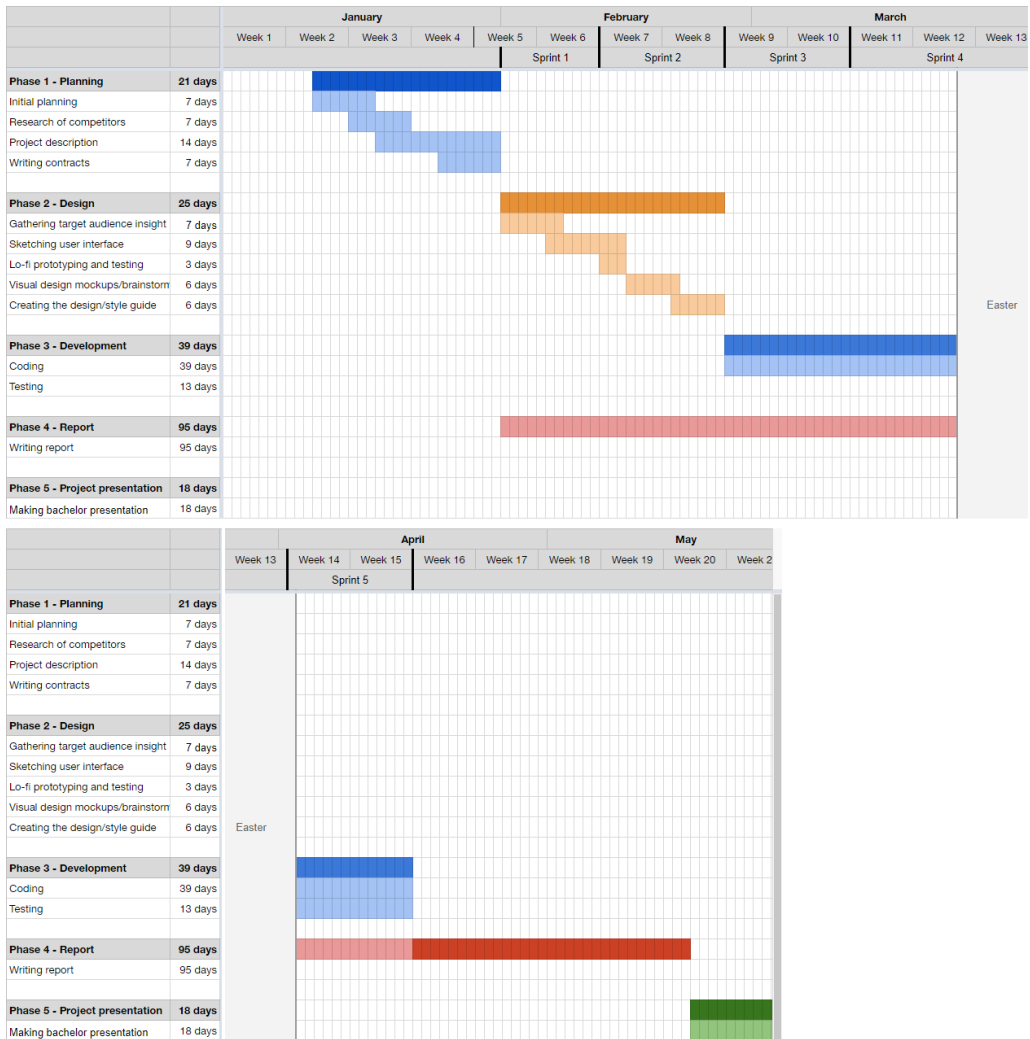
Håkon Horvei

Christoffer Seljehaug

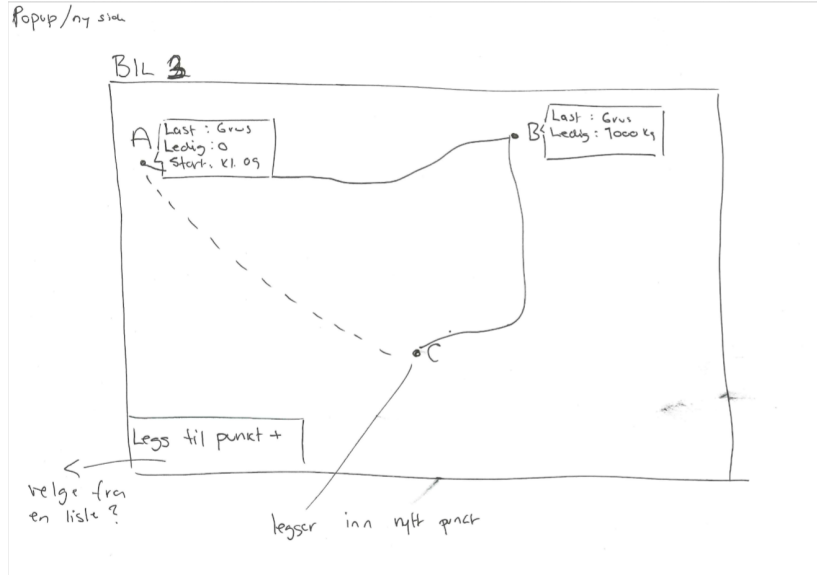
Mats Åsbu Jacobsen

Date

L Gantt chart



M Sketches



timelog

N Hour log

Name	Date	Start time	End time	Hours	Description
Aslak	11.01.18	12:00	15:00	3,00	Initial planning
Håkon	11.01.18	12:00	15:00	3,00	Meeting
Christoffer	11.01.18	12:00	15:00	3,00	Initial planning
Mats	11.01.18	12:10	15:00	2,83	Initial planning
Mats	12.01.18	14:10	17:15	3,08	Meetings with product owner and supervisor
Christoffer	12.01.18	14:00	17:15	3,25	Meeting with product owner and supervisor
Håkon	12.01.18	14:00	17:15	3,25	Meeting with supervisor and product owner.
Aslak	12.01.18	14:00	17:15	3,25	Meeting with product owner and supervisor
Mats	15.01.18	10:15	16:00	5,75	Project brainstorming
Christoffer	15.01.18	10:00	16:00	6,00	Project brainstorming
Aslak	15.01.18	18:00	20:00	2,00	Researching other solutions
Håkon	15.01.18	15:00	18:00	3,00	Rival research
Mats	17.01.18	11:00	16:05	5,08	Competitor research/analysis and general planning
Christoffer	17.01.18	10:00	16:00	6,00	Planning, Research, Project discussion
Aslak	17.01.18	10:00	16:00	6,00	Planning, project discussion, research
Håkon	17.01.18	10:00	16:00	6,00	Planning, project discussion, research
Christoffer	18.01.18	14:00	16:00	2,00	Project plan
Aslak	18.01.18	16:00	17:15	1,25	Writing logs, and setting up for meeting summaries
Mats	19.01.18	09:00	15:00	6,00	Meeting with supervisor, project planning, and sending emails
Aslak	19.01.18	09:00	15:00	6,00	Meeting, project plan, email, summaries
Håkon	19.01.18	09:00	15:00	6,00	Meeting, project plan, emails, summaries.
Mats	24.01.18	14:40	16:00	1,33	Meeting with product owner
Christoffer	24.01.18	18:00	19:00	1,00	Project report - goals/objectives
Aslak	24.01.18	10:00	16:00	6,00	Internal contract and prep for skype meeting with product owner. Meeting with product owner
Håkon	24.01.18	10:00	16:00	6,00	Planning and brainstorming
Mats	26.01.18	10:10	17:10	7,00	Started creating the Gantt schema, and worked on creating the project contract
Christoffer	26.01.18	10:00	17:00	7,00	Writing project contracts + work on Gantt chart
Aslak	26.01.18	10:00	17:00	7,00	External contract. Gantt schema.
Mats	29.01.18	10:20	14:00	3,67	Gantt schema and working on contracts
Christoffer	29.01.18	10:00	14:00	4,00	Contracts, Gantt
Christoffer	29.01.18	16:00	18:00	2,00	Contracts, Gantt
Mats	29.01.18	16:00	18:00	2,00	Finished gantt schema and continued working on contracts
Håkon	29.01.18	10:00	14:00	4,00	Work
Aslak	29.01.18	10:00	14:00	4,00	Gantt schema
Håkon	29.01.18	16:00	18:00	2,00	More work
Aslak	29.01.18	16:00	18:00	2,00	Gantt schema and contract
Mats	30.01.18	10:15	15:30	5,25	Worked on project plan documents and finished up contracts
Christoffer	30.01.18	10:00	15:30	5,50	Project plan, contracts
Håkon	30.01.18	10:00	15:00	5,00	Project plan, contracts, GANTT ++
Aslak	30.01.18	10:00	15:30	5,50	Project plan, contracts
Christoffer	31.01.18	10:00	16:00	6,00	Project plan
Aslak	31.01.18	10:15	16:00	5,75	Signing contracts, project plan
Mats	31.01.18	11:30	16:00	4,50	Signed contracts and wrote on project plan
Håkon	01.02.18	10:00	15:00	5,00	Project plan
Christoffer	01.02.18	10:15	14:30	4,25	Finishing contract plan, making contact with companies
Aslak	01.02.18	10:15	14:30	4,25	Finishing touches on project plan. Emails to companies.

Name	Date	Start time	End time	Hours	Description
Mats	01.02.18	10:20	14:30	4,17	Finished project plan and contacted transportation companies
Aslak	05.02.18	10:00	16:00	6,00	Making interview/survey questions and sending emails to companies
Christoffer	05.02.18	10:15	16:00	5,75	Making contact with companies, making interview and survey questions
Mats	05.02.18	10:30	16:00	5,50	Created interview guide, created and sent out survey
Håkon	05.02.18	10:00	16:00	6,00	interview and survey
Mats	07.02.18	10:25	14:35	4,17	Sketching desktop user interface
Christoffer	07.02.18	10:00	14:35	4,58	Sketching desktop user interface (for company)
Håkon	07.02.18	10:00	14:30	4,50	Sketching
Mats	08.02.18	10:15	15:25	5,17	Meeting with product owner
Christoffer	08.02.18	10:15	15:25	5,17	Meeting with product owner and supervisor
Håkon	08.02.18	10:15	15:15	5,00	Meeting product owner and bachelor supervisor
Håkon	08.02.18	10:15	15:25	5,17	Meeting with product owner
Christoffer	09.02.18	10:00	13:00	3,00	Conducting interview with company
Mats	09.02.18	16:10	18:25	2,25	Transcribing interview.
Aslak	09.02.18	10:00	13:00	3,00	Interview
Håkon	09.02.18	10:00	13:00	3,00	Conducting interview
Mats	10.02.18	16:20	18:30	2,17	Transcribing interview
Mats	10.02.18	19:55	21:30	1,58	Transcribing interview
Mats	13.02.18	12:30	16:00	3,50	Conducting interview
Aslak	13.02.18	12:50	15:50	3,00	Interview
Håkon	13.02.18	13:50	15:50	2,00	Conducting interview
Christoffer	13.02.18	12:30	16:00	3,50	Conducting interview with company
Christoffer	15.02.18	10:30	16:00	5,50	Sketching
Aslak	15.02.18	10:30	16:00	5,50	Sketching
Aslak	16.02.18	09:00	14:00	5,00	Guidance, and some discussing and writing after
Christoffer	16.02.18	09:00	14:00	5,00	Guidance meeting with supervisor + discussion/writing afterwards
Håkon	16.02.18	09:00	14:00	5,00	Guidance meeting and discussion
Aslak	19.02.18	10:00	15:00	5,00	Writing about scope and reasonings for our project constrictions
Christoffer	19.02.18	10:00	15:00	5,00	Writing about scope and reasonings for our project constrictions
Mats	19.02.18	10:00	15:00	5,00	Writing about scope and reasonings for our project constrictions
Håkon	19.02.18	10:00	15:00	5,00	Writing about scope and reasonings for our project constrictions
Aslak	20.02.18	10:00	14:00	4,00	Writing about scope and reasonings for our project constrictions
Christoffer	20.02.18	10:00	14:00	4,00	Writing about scope and reasonings for our project constrictions
Mats	20.02.18	10:00	14:00	4,00	Writing about scope and reasonings for our project constrictions
Aslak	21.02.18	10:00	15:00	5,00	Writing about scope and reasonings for our project constrictions
Christoffer	21.02.18	10:00	15:00	5,00	Writing about scope and reasonings for our project constrictions
Mats	21.02.18	10:00	15:00	5,00	Writing about scope and reasonings for our project constrictions
Aslak	26.02.18	12:00	15:00	3,00	Research of adobe XD
Mats	26.02.18	12:00	15:00	3,00	Research of adobe XD

Name	Date	Start time	End time	Hours	Description
Christoffer	05.03.18	10:00	14:00	4,00	Working on digital mockups
Aslak	05.03.18	10:00	14:00	4,00	Digital Mockup
Mats	05.03.18	10:00	14:00	4,00	Digital mockups / design
Christoffer	06.03.18	14:00	17:00	3,00	Started transcribing interview 2
Christoffer	07.03.18	13:00	18:00	5,00	Discussion + working on digital mockups
Aslak	07.03.18	13:00	18:00	5,00	Discussion about digital mockups
Mats	07.03.18	13:00	18:00	5,00	Design and discussions
Håkon	07.03.18	13:00	18:00	5,00	Prototyping and discussions.
Christoffer	08.03.18	10:30	17:00	6,50	Discussion + working on digital mockups
Aslak	08.03.18	10:30	17:00	6,50	Discussion about digital mockup
Mats	08.03.18	10:30	17:00	6,50	Design and discussions
Håkon	08.03.18	10:30	17:00	6,50	Prototyping and discussions.
Christoffer	09.03.18	12:00	15:30	3,50	Discussion + working on digital mockups
Aslak	09.03.18	12:00	15:30	3,50	Discussion about digital mockup
Mats	09.03.18	12:00	15:30	3,50	Design and discussions
Håkon	09.03.18	12:00	15:30	3,50	Prototyping and discussions.
Christoffer	12.03.18	10:30	17:00	6,50	Discussion + working on digital mockups
Mats	12.03.18	11:00	17:00	6,00	Design and discussions
Aslak	12.03.18	10:30	17:00	6,50	Discussion about digital mockup
Håkon	12.03.18	10:30	17:00	6,50	Prototyping and discussions.
Mats	13.03.18	10:20	16:00	5,67	Design and discussions
Aslak	13.03.18	10:00	16:00	6,00	Discussion about digital mockup
Håkon	13.03.18	10:00	16:00	6,00	Prototyping and discussions.
Mats	15.03.18	10:40	14:30	3,83	Design and discussions
Christoffer	15.03.18	10:30	14:30	4,00	Design and discussions
Aslak	15.03.18	10:30	14:30	4,00	Digital prototyping and discussions
Håkon	15.03.18	10:30	14:30	4,00	Prototyping and discussions.
Mats	16.03.18	11:45	17:00	5,25	Design and discussions
Christoffer	16.03.18	10:30	17:00	6,50	Design and discussions
Aslak	16.03.18	11:00	17:00	6,00	Digital prototyping and discussions
Håkon	16.03.18	10:30	17:00	6,50	Prototyping and discussions.
Mats	20.03.18	09:10	17:00	7,83	Design and discussions
Christoffer	20.03.18	09:00	17:00	8,00	Design and discussion
Aslak	20.03.18	09:30	17:00	7,50	Digital prototyping and discussions
Håkon	20.03.18	09:00	17:00	8,00	Prototyping and discussions.
Mats	21.03.18	10:30	19:05	8,58	Finished digital prototype
Aslak	21.03.18	10:15	19:05	8,83	Finishing digital prototype
Christoffer	22.03.18	10:30	16:30	6,00	Planning next sprint + User testing + meeting with product owner
Mats	22.03.18	10:30	16:30	6,00	Planning next sprint + User testing + meeting with product owner
Håkon	22.03.18	10:30	16:30	6,00	Meeting and usertests
Christoffer	23.03.18	08:00	15:00	7,00	Meeting with supervisor, discussion, user testing
Mats	23.03.18	08:20	15:00	6,67	Meeting with supervisor, user testing, discussed future plans/backlog
Håkon	23.03.18	08:00	15:00	7,00	Meeting with supervisor and conducting user testing
Håkon	26.03.18	10:00	18:00	8,00	Angular courses, research and testing.
Christoffer	27.03.18	16:00	19:15	3,25	Researching Google Maps API in combination with Angular
Håkon	27.03.18	12:00	17:00	5,00	Angular courses, research and testing.
Håkon	29.03.18	10:00	18:00	8,00	Angular courses, research and testing.
Håkon	30.03.18	12:00	18:00	6,00	Angular courses, research and testing.

Name	Date	Start time	End time	Hours	Description
Christoffer	02.04.18	13:00	17:00	4,00	Testing out Google Maps API in Angular
Aslak	03.04.18	14:00	15:30	1,50	Researching Angular
Christoffer	04.04.18	12:00	18:30	6,50	Angular Google Maps API
Mats	04.04.18	15:05	19:15	4,17	Angular testing/research
Mats	04.04.18	22:40	23:45	1,08	Angular Google Maps API
Aslak	04.04.18	12:00	16:00	4,00	Researching Angular
Christoffer	05.04.18	11:30	18:00	6,50	Angular Google Maps API
Mats	05.04.18	14:00	17:40	3,67	Angular Google Maps API
Mats	05.04.18	19:45	23:05	3,33	Angular Google Maps API
Aslak	05.04.18	11:00	15:00	4,00	Writing and organizing documents
Håkon	05.04.18	16:00	20:00	4,00	Setting up project in GitHub, implementing structure for SASS.
Christoffer	06.04.18	12:00	15:00	3,00	Angular google maps
Mats	06.04.18	12:15	14:45	2,50	Meeting about the plan forward + Angular Google Maps API
Aslak	06.04.18	12:00	15:00	3,00	Meeting to discuss way forward
Håkon	06.04.18	12:00	14:00	2,00	Enhancing structure of project
Mats	07.04.18	13:35	15:30	1,92	Angular Google Maps API
Christoffer	07.04.18	15:00	16:30	1,50	Angular autocomplete
Christoffer	07.04.18	18:20	19:55	1,58	Angular autocomplete
Mats	07.04.18	18:35	21:10	2,58	Angular Google Maps API
Mats	08.04.18	14:20	15:40	1,33	Angular Google Maps API
Christoffer	08.04.18	16:00	21:05	5,08	Angular Google Maps
Mats	08.04.18	19:30	23:10	3,67	Angular Google Maps API
Aslak	08.04.18	12:00	15:00	3,00	Researching angular, writing on report
Håkon	08.04.18	12:00	18:00	6,00	Angular research, testing buildmode
Christoffer	09.04.18	12:10	19:00	6,83	Google Maps API
Mats	09.04.18	12:10	19:00	6,83	Angular Google Maps API
Håkon	09.04.18	18:00	21:00	3,00	Angular research and small update on project
Christoffer	10.04.18	12:30	18:00	5,50	Angular Google Maps
Mats	10.04.18	12:30	18:00	5,50	Angular Google Maps
Håkon	10.04.18	10:00	14:00	4,00	Making navigation in Angular
Håkon	10.04.18	18:00	23:00	5,00	Making navigation in Angular
Christoffer	11.04.18	12:30	18:00	5,50	Angular Google Maps API
Mats	11.04.18	12:30	18:00	5,50	Angular Google Maps
Håkon	11.04.18	11:00	20:00	9,00	Angular layout
Håkon	11.04.18	22:00	23:55	1,92	Updating sidebar
Christoffer	12.04.18	12:30	18:00	5,50	Angular Google Maps API
Mats	12.04.18	12:30	18:00	5,50	Angular Google Maps
Håkon	12.04.18	11:00	18:00	7,00	Researching and testing splitting of map and table
Christoffer	13.04.18	13:00	14:50	1,83	Angular Google Maps API
Christoffer	13.04.18	16:20	18:35	2,25	Angular Google Maps API
Håkon	13.04.18	09:00	16:00	7,00	Implementing splitting, testing and fixing bugs
Christoffer	14.04.18	18:00	20:10	2,17	Angular Google Maps API
Mats	14.04.18	15:00	16:30	1,50	Angular Google Maps, RouteBoxer
Mats	14.04.18	19:10	21:20	2,17	Angular Google Maps, RouteBoxer
Christoffer	15.04.18	14:30	19:20	4,83	Angular Google Maps API + Angular Datatables
Christoffer	15.04.18	21:25	22:35	1,17	Angular Datatables
Mats	15.04.18	16:50	19:30	2,67	Angular Google Maps
Mats	15.04.18	19:55	23:05	3,17	Angular Google Maps
Mats	15.04.18	02:00	03:30	1,50	Angular Google Maps
Mats	16.04.18	12:00	16:00	4,00	Angular Google Maps, merged files to main project
Christoffer	16.04.18	10:30	16:00	5,50	Testing material datatables in angular

Name	Date	Start time	End time	Hours	Description
Mats	16.04.18	18:00	21:05	3,08	Research + Angular Google Maps
Christoffer	16.04.18	19:00	21:50	2,83	Angular Material table
Aslak	16.04.18	10:00	12:00	2,00	Inserting dummy data into database setup file
Christoffer	17.04.18	11:30	13:10	1,67	Angular Material table
Christoffer	17.04.18	09:00	10:30	1,50	Meeting with supervisor and some discussion after
Mats	17.04.18	13:15	15:05	1,83	Angular Google Maps
Mats	17.04.18	09:00	10:30	1,50	Meeting with supervisor + Angular Google Maps
Christoffer	17.04.18	15:00	20:45	5,75	Angular Material table
Mats	17.04.18	17:50	20:50	3,00	Angular Google Maps
Aslak	17.04.18	12:00	15:00	3,00	Meeting with supervisor and report
Mats	18.04.18	00:00	01:05	1,08	Angular Google Maps
Christoffer	18.04.18	11:30	15:00	3,50	Angular Material table
Christoffer	18.04.18	15:45	17:45	2,00	Angular Material table
Mats	18.04.18	15:00	18:00	3,00	Angular Google Maps
Christoffer	18.04.18	19:00	21:25	2,42	Angular Material table
Mats	18.04.18	19:00	22:40	3,67	Angular Google Maps, Photoshop
Christoffer	19.04.18	11:00	14:00	3,00	Angular Material table
Christoffer	19.04.18	15:00	16:55	1,92	Angular Material table
Mats	19.04.18	15:15	19:20	4,08	Angular database connection and design
Christoffer	19.04.18	19:00	22:20	3,33	Angular Material table
Mats	19.04.18	20:45	23:55	3,17	Angular database connection
Aslak	19.04.18	12:00	16:30	4,50	Configuring latex template
Christoffer	20.04.18	11:15	15:20	4,08	Angular Material table
Christoffer	20.04.18	17:00	20:50	3,83	Angular Material table, swapping out JSON with DB data
Aslak	20.04.18	13:00	14:45	1,75	Finishing config on template
Håkon	20.04.18	10:00	18:00	8,00	Updating navigation and sidebar
Håkon	20.04.18	19:00	23:55	4,92	Datepickers
Christoffer	21.04.18	12:00	13:25	1,42	Angular Material table
Christoffer	21.04.18	20:30	22:00	1,50	Angular Material table
Mats	21.04.18	15:30	19:40	4,17	Angular Google Maps and layout
Christoffer	21.04.18	15:35	19:50	4,25	Angular Material table
Aslak	21.04.18	14:00	17:00	3,00	report
Håkon	21.04.18	11:00	18:00	7,00	Formating and transferring data
Håkon	21.04.18	19:00	23:55	4,92	Filtering in application
Mats	22.04.18	00:00	01:30	1,50	Angular
Christoffer	22.04.18	13:40	14:45	1,08	Angular Material table
Mats	22.04.18	20:10	23:10	3,00	Angular Google Maps
Mats	22.04.18	15:20	19:45	4,42	Angular Google Maps
Christoffer	22.04.18	15:40	20:40	5,00	Angular Material table
Aslak	22.04.18	16:00	18:40	2,67	Report
Mats	23.04.18	12:30	16:30	4,00	Discussing disposition of report
Aslak	23.04.18	12:30	16:20	3,83	Discussing disposition of report
Christoffer	23.04.18	12:30	16:25	3,92	Disussing project report disposition
Håkon	23.04.18	18:00	23:55	5,92	Styling and bugfixing
Håkon	23.04.18	12:00	16:30	4,50	Discussing report disposition
Christoffer	24.04.18	11:00	15:00	4,00	Meeting with supervisor, planning project disposition
Mats	24.04.18	11:00	15:05	4,08	Meeting with supervisor, and worked on report disposition
Christoffer	24.04.18	16:00	18:35	2,58	Report
Mats	24.04.18	18:35	21:05	2,50	Report
Mats	24.04.18	22:35	23:00	0,42	Report

Name	Date	Start time	End time	Hours	Description
Christoffer	24.04.18	19:30	20:10	0,67	Report
Aslak	24.04.18	11:00	15:00	4,00	Meeting with supervisor, report disposition
Aslak	24.04.18	16:15	18:00	1,75	Report
Håkon	24.04.18	11:00	15:00	4,00	Meeting supervisor about report disposition
Christoffer	25.04.18	10:20	11:20	1,00	Report
Christoffer	25.04.18	13:30	16:40	3,17	Report
Mats	25.04.18	15:40	17:35	1,92	Report
Christoffer	25.04.18	17:30	18:40	1,17	Report
Mats	25.04.18	20:00	22:45	2,75	Report
Aslak	25.04.18	11:30	15:00	3,50	Report
Aslak	25.04.18	16:30	18:00	1,50	Report
Christoffer	26.04.18	10:40	12:20	1,67	Report
Christoffer	26.04.18	13:30	16:30	3,00	Report
Aslak	26.04.18	12:00	16:00	4,00	Report
Christoffer	26.04.18	19:20	20:40	1,33	Report
Mats	26.04.18	20:10	22:15	2,08	Report/Research
Christoffer	27.04.18	10:50	12:30	1,67	Report
Aslak	27.04.18	11:00	14:00	3,00	report
Christoffer	27.04.18	13:55	17:00	3,08	Report
Christoffer	28.04.18	11:15	13:15	2,00	Report
Christoffer	28.04.18	15:30	17:25	1,92	Report
Mats	28.04.18	19:00	20:35	1,58	Report
Mats	28.04.18	21:50	23:00	1,17	Report
Mats	29.04.18	00:00	01:20	1,33	Report
Christoffer	04.05.18	18:00	19:15	1,25	styles and select list in table
Christoffer	04.05.18	19:15	20:05	0,83	Report
Christoffer	05.05.18	11:15	12:25	1,17	Report
Christoffer	05.05.18	14:00	15:20	1,33	Report
Christoffer	06.05.18	16:20	17:20	1,00	Report
Christoffer	07.05.18	10:20	15:25	5,08	Report
Mats	07.05.18	14:00	15:45	1,75	Report
Aslak	07.05.18	10:05	15:25	5,33	Report
Christoffer	07.05.18	17:00	19:00	2,00	Report
Aslak	07.05.18	17:00	19:00	2,00	Report
Mats	07.05.18	20:40	22:05	1,42	Report
Håkon	07.05.18	10:00	18:00	8,00	Report
Aslak	08.05.18	10:15	15:00	4,75	Report
Christoffer	08.05.18	10:15	15:00	4,75	Report
Christoffer	08.05.18	15:50	17:55	2,08	Report
Aslak	08.05.18	15:50	18:25	2,58	Report
Mats	08.05.18	00:10	00:40	0,50	Report
Mats	08.05.18	15:00	17:15	2,25	Report
Mats	08.05.18	17:50	20:55	3,08	Report
Håkon	08.05.18	10:00	18:00	8,00	Report
Christoffer	09.05.18	09:00	12:00	3,00	Meeting with supervisor, report
Mats	09.05.18	09:00	12:00	3,00	Meeting with supervisor and group discussion about the report
Christoffer	09.05.18	13:45	16:30	2,75	Report
Mats	09.05.18	17:45	20:15	2,50	Report
Christoffer	09.05.18	19:00	20:50	1,83	Report
Aslak	09.05.18	09:00	14:20	5,33	Report
Aslak	09.05.18	16:30	17:45	1,25	Report
Håkon	09.05.18	09:00	18:00	9,00	Meeting + Report

Name	Date	Start time	End time	Hours	Description
Mats	10.05.18	00:20	01:00	0,67	Report
Christoffer	10.05.18	10:30	12:20	1,83	Report
Christoffer	10.05.18	14:00	16:40	2,67	Report
Aslak	10.05.18	13:45	15:50	2,08	Report
Christoffer	10.05.18	19:10	21:25	2,25	Report
Aslak	10.05.18	16:30	17:40	1,17	Report
Mats	10.05.18	15:00	16:10	1,17	Report
Mats	10.05.18	18:30	21:50	3,33	Report
Håkon	10.05.18	10:00	18:00	8,00	Report
Håkon	10.05.18	20:00	22:00	2,00	Report
Christoffer	11.05.18	10:10	14:30	4,33	Report
Aslak	11.05.18	18:00	20:10	2,17	Report
Aslak	11.05.18	10:10	15:30	5,33	Report
Christoffer	11.05.18	17:10	20:30	3,33	Report
Håkon	11.05.18	10:00	18:00	8,00	Report
Mats	11.05.18	12:30	15:35	3,08	Report
Mats	11.05.18	16:10	20:30	4,33	Report
Mats	11.05.18	22:00	22:45	0,75	Report
Christoffer	12.05.18	12:00	16:15	4,25	Report
Christoffer	12.05.18	18:30	21:15	2,75	Report
Aslak	12.05.18	14:45	17:00	2,25	Report
Aslak	12.05.18	18:15	21:40	3,42	Report
Mats	12.05.18	00:30	02:30	2,00	Report
Håkon	12.05.18	10:00	18:00	8,00	Report
Mats	12.05.18	14:15	16:35	2,33	Report
Mats	12.05.18	18:20	21:55	3,58	Report
Christoffer	13.05.18	11:05	18:55	7,83	Report
Mats	13.05.18	11:40	19:00	7,33	Report
Christoffer	13.05.18	20:00	20:40	0,67	Report
Aslak	13.05.18	11:05	18:55	7,83	Report
Håkon	13.05.18	11:00	16:00	5,00	Report
Håkon	13.05.18	18:00	21:00	3,00	Report
Christoffer	14.05.18	10:20	16:00	5,67	Report
Christoffer	14.05.18	17:00	19:00	2,00	Report
Mats	14.05.18	11:45	16:00	4,25	Report
Mats	14.05.18	17:00	19:45	2,75	Report
Christoffer	14.05.18	19:40	21:00	1,33	Report
Aslak	14.05.18	10:30	16:00	5,50	Report
Aslak	14.05.18	17:00	20:40	3,67	Report
Håkon	14.05.18	11:00	16:00	5,00	Report
Håkon	14.05.18	20:00	23:55	3,92	Report
Christoffer	15.05.18	11:00	23:55	12,92	Report
Aslak	15.05.18	11:00	23:55	12,92	report
Håkon	15.05.18	11:15	23:55	12,67	Final work on report
Mats	15.05.18	11:00	23:55	12,92	Report
Christoffer	16.05.18	00:00	10:00	10,00	Report
Aslak	16.05.18	00:00	10:00	10,00	report
Håkon	16.05.18	00:00	10:00	10,00	Final work on report
Mats	16.05.18	00:00	10:00	10,00	Report

Planlagte oppdrags oversikt

14.02.18	Gjøvik	Lillehammer	50 min 20 min grus	4 (Volvo XS20Z)	Emat Nilsen
Sejl info					
Beregnet tid: 50 min					
50% av ruten kjøres med tomlast					
2 oppdrag i nærheten av ruten legger ute <input type="checkbox"/> vis					
som kan redusere tomlast spesiell til 25%					
(KART)					
14.02.18	Gjøvik	Hamar	1 tonn seil	2 (Honda SLX)	Sumat Olsen
vis info					

Startpunkt Destinasjon Legg inn punkt

på karte, les til i kart- og rett input field

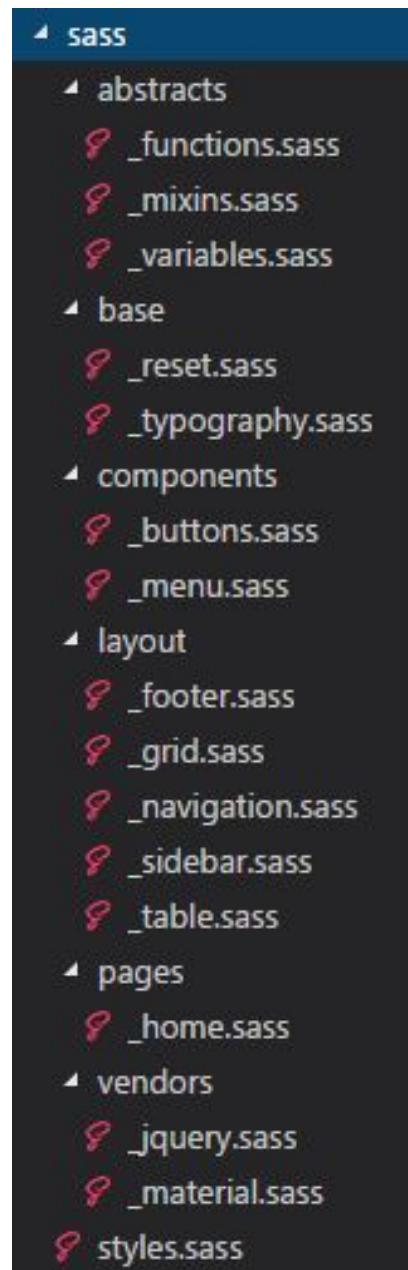
Original
Forslag
Bud
Rerouter

filtrer etter

Fra J til K

Gå til oppdrag i stedet?

O Sass structure



P Code snippets for table

P.1 JobListingsDatabase

```
1 // Gets data from the database service. This data is later used in the data source
2 export class JobListingsDatabase {
3
4   jobListings = new BehaviorSubject([]);
5   get data() { return this.jobListings.value; }
6
7   constructor(private tableService: TableService, private databaseService: DatabaseService,
8     private mapsService: MapService) {
9     // Get the data for the table rows
10    this.databaseService.fetchDeliveryJobs().subscribe(
11      response => {
12        let rowData = [];
13        let deliveryRoutes = [];
14        // Build delivery routes with correct format
15        response['data'].forEach(row => {
16          rowData.push(row);
17
18          deliveryRoutes.push({
19            origin: {
20              lat: Number(row.start_lat),
21              lng: Number(row.start_lng)
22            },
23            destination: {
24              lat: Number(row.end_lat),
25              lng: Number(row.end_lng)
26            },
27            id: Number(row.assignment_id),
28            visible: true
29          });
30          this.jobListings.next(rowData); // update the jobListings
31          this.mapsService.setDeliveryJobRoutes(deliveryRoutes); // update delivery job
32          routes
33        },
34        error => {
35          alert('Could not connect to database.');
```

P.2 JobListingsDataSource

```

1 // Provides the data to be rendered in the table.
2 export class JobListingsDataSource extends DataSource<any> {
3   constructor(private jobListingsDatabase: JobListingsDatabase, private sort: MatSort) {
4     super();
5   }
6
7 // Return an observable with the exact data to be rendered in the table
8 connect(): Observable<TableRow[]> {
9   const jobListingsData = [
10    this.jobListingsDatabase.jobListings,
11    this.sort.sortChange
12  ];
13
14  return Observable.merge(...jobListingsData).map(() => {
15    return this.getSortedData();
16  });
17 }
18
19 disconnect() { }
20
21 // Code from Material Table used for sorting by table columns
22 getSortedData() {
23   const data = this.jobListingsDatabase.data.slice();
24   if (!this.sort.active || this.sort.direction === '') { return data; }
25
26   return data.sort((a, b) => {
27     let propertyA: number | string = '';
28     let propertyB: number | string = '';
29
30     switch (this.sort.active) {
31       case 'entrepreneurName': [propertyA, propertyB]=[a.entr_name, b.entr_name]; break;
32       case 'pickupLocation': [propertyA, propertyB]=[a.start_city, b.start_city]; break;
33       case 'dropoffLocation': [propertyA, propertyB]=[a.end_city, b.end_city]; break;
34       case 'transportationWindow': [propertyA, propertyB]=[a.window_start, b.window_start
35 ]; break;
36       // case 'cargo': [propertyA, propertyB]=[a.short_desc, b.short_desc]; break;
37       case 'weight': [propertyA, propertyB]=[a.weight, b.weight]; break;
38       // case 'dimensions': [propertyA, propertyB]=[a.dimensions, b.dimensions]; break;
39       case 'lowestBid': [propertyA, propertyB]=[a.lowest_bid, b.lowest_bid]; break;
40       case 'bidDeadline': [propertyA, propertyB]=[a.bid_deadline, b.bid_deadline]; break;
41     }
42     let valueA = isNaN(+propertyA) ? propertyA : +propertyA;
43     let valueB = isNaN(+propertyB) ? propertyB : +propertyB;
44
45     return (valueA < valueB ? -1 : 1) * (this.sort.direction === 'asc' ? 1 : -1);
46   });
47 }

```

Q Digital prototype

BACKLOAD Oppdrag Min oversikt

Legg inn rute

Dato

Kjørerute

Last

- STYKK- OG PARTIGODS
- LØSMASSE
- ANNET

Vis på kart	OPPRAGSGIVER	HENTESTED	LEVERINGSSTED	TRANSPORTVINDU	LAST	VEKT	LENGDE, BREDD, HØYDE	LAIVESTE BUD	BUDFRIST
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00

Figure 1: Before you have input date or route

BACKLOAD Oppdrag Min oversikt

Legg inn rute

Dato

17. mars - 19. mars

Kjørerute

- Gjenvik
- SKJUL DELROUTE
- Lillemhammer
- SKJUL DELROUTE
- Gjenvik
- Nytt punkt

Last

- STYKK- OG PARTIGODS
- LØSMASSE
- ANNET

Vis på kart	OPPRAGSGIVER	HENTESTED	LEVERINGSSTED	TRANSPORTVINDU	LAST	VEKT	LENGDE, BREDD, HØYDE	LAIVESTE BUD	BUDFRIST
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00
<input type="radio"/>	Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Ståløyelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00

Figure 2: After input is done, shows a horizontal split of table and map

BACKLOAD Oppdrag Min oversikt

Legg inn rute

Dato: 17. mars - 19. mars

Kjørerute

- Gjøvik
- Lillehammer
- Gjøvik
- Nytt punkt

Last

- STYKK- OG PARTIGODS
- LØSMASSE
- ANNET

OPPDRAGSGIVER	HENTESTED	LEVERINGSSTED	TRANSPORTVINDU	LAST	VEKT	LENGDE, BREDD, HOYDE	LAVESTE BUD	BUDEFRIST
Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Stålbjelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00

15 stålbjelker pakket 3x5

Vekt: 850kg
Lengde, bredde, høyde: 250x200x50m
Laveste bud: 7800kr

HENTESTED: Rogalandsgata 42, 4401 Stange
LEVERINGSSTED: Lundsgate 13, 2815 Sletta

BUDEFRIST	BUDGIVER	BUD
15min	Erik & Eriksen Godstransport (2)	8700kr
40min	Toten Transport AS (14)	9000kr
40min	Toten Transport AS (14)	9000kr

Ansatt hentetid: 17. Mars 13:00 - Ansatt leveringstid: 17. Mars 13:00 -

Budpris: kr

Figure 5: Only table with a table element clicked on

BACKLOAD Oppdrag Min oversikt

Aktive oppdrag

Budrunder

Lagrede oppdrag

Historikk

Stål & Spesialtransport AS Telefon: 90141422 E-post: kontakt@staaltransport.no

Stålbjelker 250x200x50 cm, 850kg

HENTESTED: Rogalandsgata 42, 4401 Stange 15. Mars 14:00
LEVERINGSSTED: Lundsgate 13, 2815 Sletta 16. Mars 11:00

Stål & Spesialtransport AS Telefon: 90141422 E-post: kontakt@staaltransport.no

Stålbjelker 250x200x50 cm, 850kg

HENTESTED: Rogalandsgata 42, 4401 Stange 15. Mars 14:00
LEVERINGSSTED: Lundsgate 13, 2815 Sletta 16. Mars 11:00

Stål & Spesialtransport AS Telefon: 90141422 E-post: kontakt@staaltransport.no

Stålbjelker 250x200x50 cm, 850kg

HENTESTED: Rogalandsgata 42, 4401 Stange
LEVERINGSSTED: Lundsgate 13, 2815 Sletta

Figure 6: Users active transportation assignments

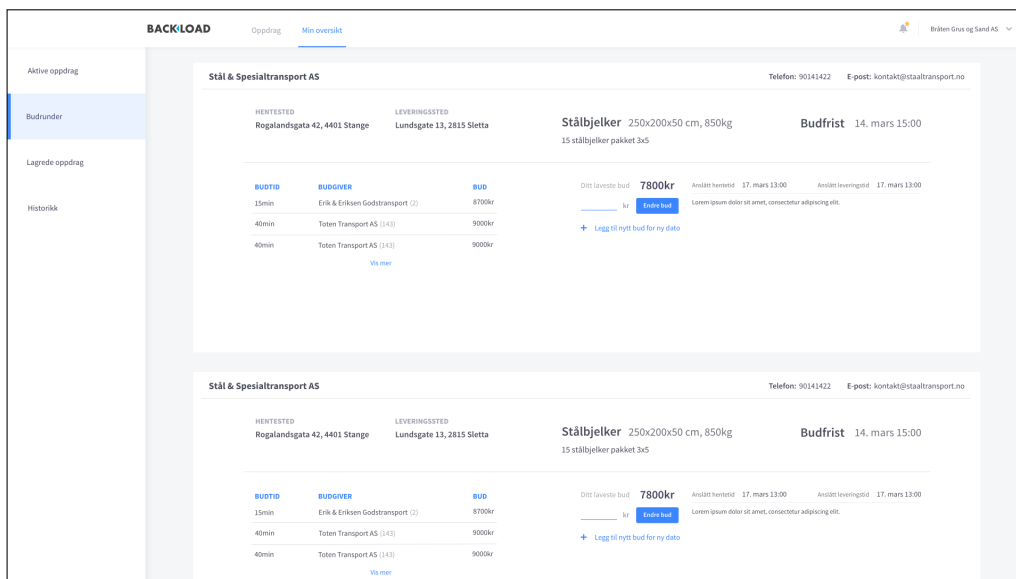


Figure 7: Users active bidding wars

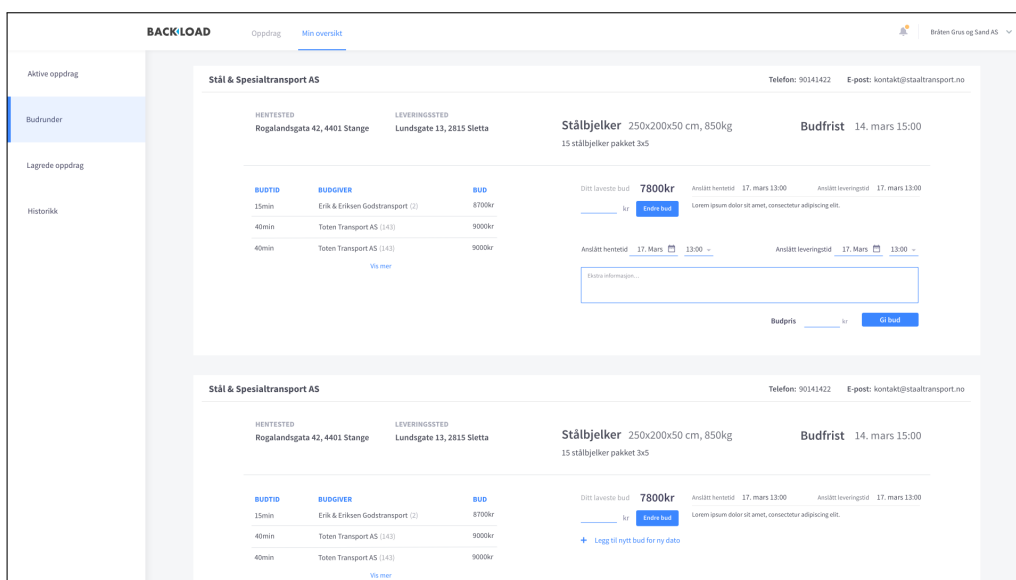


Figure 8: Input for adding a separate bid with a new date

The screenshot shows two identical bid detail panels. Each panel is for 'Stål & Spesialtransport AS' with contact info: Telefon: 90141422, E-post: kontakt@staaltransport.no. The origin is Rogalandsgata 42, 4401 Stange and the destination is Lundgate 13, 2815 Sletta. The item is 'Stålbjelker 250x200x50 cm, 850kg' with a deadline of 14. mars 15:00. A table lists competitors and their bids:

BUDTID	BUDGIVER	BUD
15min	Erik & Eriksen Godstransport (2)	8700kr
40min	Toten Transport AS (143)	9000kr
40min	Toten Transport AS (143)	9000kr

Below the table, the user's bid is shown as 7800kr with a 'Endre bud' button. A second bid of 7700kr is also visible below it.

Figure 9: Bid with new date added, both can be updated separately

The screenshot shows the 'Min oversikt' section of the BACKLOAD interface. It contains a table of favorite bids:

OPDRAGSGIVER	HEMTESTED	LEVERINGSSTED	TRANSPORTID	LAST	VEKT	LENGDE, BREDD, HØYDE	LAVESTE BUD	BUDFRIST	FAVORIT
Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Stålbjelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00	★
Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Stålbjelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00	★
Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Stålbjelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00	★
Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Stålbjelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00	★
Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Stålbjelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00	★
Stål & Spesialtransport AS	2815 Sletta	4401 Stange	15. Mars - 28. Mars	Stålbjelker	850kg	250x200x50 cm	7800kr	8. Mars 14:00	★

Figure 10: Users favorites