

# Design and Implementation of Customer Support Awareness Dashboard

---

Carl Rygart and Johan Westerlund

DEPARTMENT OF DESIGN SCIENCES  
LUND UNIVERSITY  
2016

MASTER THESIS

TELAVOX



# Design and Implementation of Customer Support Awareness Dashboard

Carl Rygart and Johan Westerlund



**LUND**  
UNIVERSITY

# Design and Implementation of Customer Support Awareness Dashboard

Copyright © 2016 Carl Rygart and Johan Westerlund

*Published by*

Department of Design Sciences  
Faculty of Engineering LTH, Lund University  
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design (MAMM01)  
Division: Department of Design Sciences,  
Faculty of Engineering LTH, Lund University  
Supervisor: Joakim Eriksson  
Co-supervisor: Magnus Winter at Telavox AB  
Examiner: Johanna Persson

---

# Abstract

---

Customer relation is an important factor in service minded sectors, and providing a flawless support to compliment that service can be an important factor while having customers. This puts support personnel in a crucial role to handle problems that the customer requests, but also see to it that the problems never happen. Telavox, a telecommunications company, supplies various communication services to other companies and provides this kind of both reactive and proactive support. This makes it of utmost importance to create a customer support environment that can ease this way of working and see to that nothing falls through the cracks.

At Telavox today, this work is done in an in-house developed customer relation management system, which is lacking a usable overview of today's agenda. There is an existing page, but it is missing detailed information for every specific user. The purpose of this project is to design and develop a new page that makes the work for the customer service personnel, more efficient and to create a page that enables an efficient work flow for the personnel.

This master's thesis covers the development of a new overview page in the company Telavox' own customer relation management system. The thesis goes through the several design phases, and covers aspects from ideas and sketching, low-fidelity and high-fidelity prototypes, to an evolving prototype that is put in production.

This thesis resulted in a final product that is in use for the company, and shows the importance of a thorough investigation of the domain and to keep close contact with the end user. Finally, the result is analyzed and tested by the intended users as to get an understanding of what could be improved.

**Keywords:** Telavox, interaction design, user experience, development, user-centered design, user engineering

---

# Sammanfattning

---

Kundrelation är en viktig faktor i servicerelaterade yrken. Att kunna erbjuda en utmärkt kundtjänst för den produkt man tillhandahåller kan vara en viktig faktor för att kunna behålla en kund. Detta sätter kundtjänstpersonal i en kritisk roll vid bearbetning av kundproblem, speciellt för att se till att problem inte uppstår från första början. Telavox är ett telekommunikationsföretag som tillhandahåller olika kommunikationstjänster för företag och erbjuder denna typ av support. Detta gör det viktigt att skapa en kundtjänstmiljö som underlättar arbetet och ser till så inget faller mellan stolarna.

Idag på Telavox sker kundtjänsten i deras egenutvecklade kundhanteringssystem där en användbar överblick för dagens agenda saknas. Det existerar för tillfället ett verktyg för detta behov, men som saknar mer detaljerad och specifik information för personalen. Syftet med detta projekt är att utveckla ett nytt verktyg som gör arbetet för kundtjänstpersonalen mer effektivt genom att effektivisera deras arbetsflöde.

Detta examensarbete handlar om hur ett nytt översiktsverktyg kan designas och utvecklas, och mer specifikt i Telavox eget kundhanteringssystem. Arbetet går igenom flertalet designfaser och beskriver designprocesser utifrån idé, skissning, låg- och högnaturtrogna prototyper samt en prototyp som kontinuerligt utvecklades för att sedan bli satt i produktion.

Arbetet resulterade i en slutgiltig produkt som används av företaget, men visar också vikten av en grundlig undersökning av domänen och att hålla nära kontakt med slutanvändaren. Slutligen analyserades resultatet genom tester med slutanvändaren för att få en förståelse för vad som kan vidareutvecklas.

**Nyckelord:** Telavox, interaktionsdesign, användarupplevelse, utveckling, användarcentrerad design, användarpåverkad utveckling

---

## Acknowledgements

---

We would like to thank our supervisors, Joakim Eriksson at LTH and Magnus Winter at Telavox, for their guidance and support during this project. We would also like to thank our examiner Johanna Persson at LTH who helped us with supervisor feedback during summer vacation.

Also, a big thanks to the advisor and development team at Telavox for their patience, understanding and cooperation during the thesis.

Last, we would like to thank our friends and family for their support during the thesis and all the helpful advice and hints that they provided has been invaluable and helped our project immensely.

Lund, September 2016

Carl Rygart  
Johan Westerlund

---

# Table of Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background	1
1.2	Description of Telavox AB	1
1.3	Thesis purpose, goals and questions	3
1.4	Delimitations	3
1.5	Stakeholders	4
1.6	Previous experience	5
1.7	Report structure	5
<b>2</b>	<b>Theory</b>	<b>6</b>
2.1	Data collection methods	7
2.1.1	Data collection methods introduction	7
2.1.2	Observation	7
2.1.3	Interviews	7
2.1.4	Analysis	8
2.2	Design	10
2.2.1	Design essentials	10
2.2.2	The three design phases	11
2.2.3	User in focus	13
2.3	Software development	15
2.3.1	Software development introduction	15
2.3.2	Agile in general	15
2.3.3	eXtreme programming	16
2.3.4	Company development and integration	16
2.4	Usability testing	17
2.4.1	Usability testing introduction	17
2.4.2	Scenarios and thinking aloud	18
<b>3</b>	<b>Methodology</b>	<b>19</b>
3.1	Documentation	19
3.2	Preparations	20
3.3	Design development	21

3.3.1	Design development introduction . . . . .	21
3.3.2	Investigation of current situation . . . . .	21
3.3.3	Design and implementation . . . . .	22
3.3.4	Final evaluation . . . . .	24
<b>4</b>	<b>Investigation of current situation</b> _____	<b>26</b>
4.1	Quantitative analysis . . . . .	27
4.1.1	Application . . . . .	27
4.1.2	Results . . . . .	28
4.2	Elicitation of subjects for observations and interviews . . . . .	29
4.2.1	Application . . . . .	29
4.2.2	Results . . . . .	30
4.3	Observations . . . . .	30
4.3.1	Application . . . . .	30
4.3.2	Results . . . . .	31
4.4	Interviews . . . . .	31
4.4.1	Application . . . . .	31
4.4.2	Results . . . . .	32
4.5	Conclusions . . . . .	33
4.5.1	Evaluation of quantitative data . . . . .	33
4.5.2	Affinity diagram . . . . .	34
4.5.3	User persona . . . . .	35
<b>5</b>	<b>Design and implementation</b> _____	<b>37</b>
5.1	Iteration 0: Prototype phase . . . . .	38
5.1.1	Low-fidelity prototype . . . . .	38
5.1.2	High-fidelity prototype . . . . .	41
5.2	Iteration 1: First product release . . . . .	43
5.2.1	Evaluation . . . . .	45
5.3	Iteration 2: Adding functionality and new features . . . . .	46
<b>6</b>	<b>Final evaluation</b> _____	<b>50</b>
6.1	Design analysis introduction . . . . .	50
6.1.1	Analysis of Dashboard 1.0 . . . . .	51
6.1.2	Analysis of Dashboard 2.0 . . . . .	51
6.1.3	Comparison of the two dashboards . . . . .	53
6.2	Expert evaluation: UX-team at Telavox . . . . .	54
6.3	User testing . . . . .	55
<b>7</b>	<b>Conclusion and discussion</b> _____	<b>60</b>
7.1	Conclusions . . . . .	60
7.2	Discussion . . . . .	61
7.2.1	Interview result and user personas . . . . .	61
7.2.2	Involving the other parties more often . . . . .	62
7.2.3	"The Partner look" . . . . .	62
7.2.4	The concept of proactive watch section . . . . .	63



7.2.5	Future work . . . . .	63
7.3	Final remarks . . . . .	65
<b>A</b>	<b>Interview plan(in Swedish)</b> _____	<b>68</b>
<b>B</b>	<b>User personas</b> _____	<b>71</b>
B.1	Primary user persona: Anna Almér . . . . .	71
B.2	Secondary user persona: Simon Hammarström . . . . .	71
<b>C</b>	<b>Sketch illustrations</b> _____	<b>73</b>
C.1	Paper sketches . . . . .	73
C.2	Whiteboard sketches . . . . .	86

---

## Glossary

---

- Advisor** A support employee that handles issues to a specific group of customers. The goal of an advisor is to mainly work preemptive with customer issues, but they also handles current issues that is not preemptive.
- CRM** Stands for "*Customer Relationship Management*" and in this report it is centered around the handling of customer support.
- Dashboard 1.0** The appellation in the report of the old dashboard that currently is used in Partner.
- Dashboard 2.0** The appellation in the report of the new dashboard that is being designed and developed in this master's thesis project.
- Google Analytics** A freemium web based analytics service. It tracks websites and displays the data on their web service.
- HCD** Acronym for "*Human-Centered Design*".
- ID** Acronym for Identification.
- IEEE** Acronym for "*Institute of Electrical and Electronics Engineers*".
- LTH** Abbreviation for "*Faculty of Engineering LTH at Lund University*".
- URL** Abbreviation for "*Uniform Resource Locator*". Commonly known to be a link to some page on the internet. Example: <http://telavox.se>.
- UX** Abbreviation for "*User Experience*".
- Partner** Telavox' web based CRM system.
- PBX** A business telephone system that is a multiline telephone system typically used in business environments, encompassing systems ranging from small key telephone systems to large-scale private branch exchanges.
- XP** Stands for eXtreme Programming, an agile development methodology.

# Introduction

---

The chapter introduction consists of some background information of the project, description of the company the master's thesis was done for and the purpose, goal and delimitations of the project.

## 1.1 Background

This master's thesis revolves around the design and implementation of a web based tool, called dashboard, to help improve work flow and overview for customer support at the company Telavox AB.

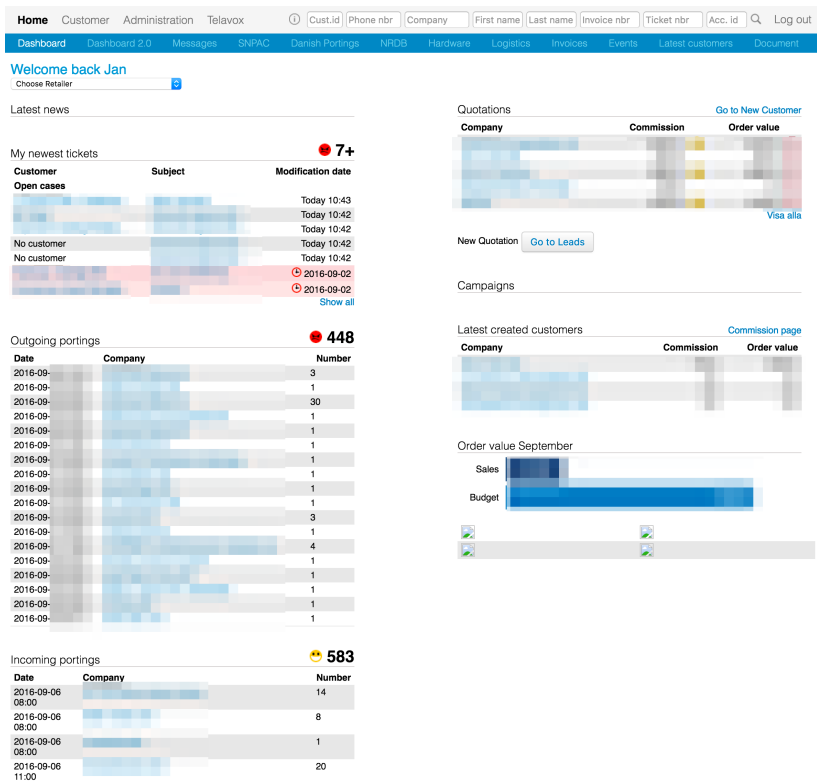
Today the customer relation management (CRM) system Partner has the page dashboard, seen in figure 1.1 with various data and issue trackers. The tool is available to all users of Partner and is the first landing page of the site. This is a tool that could potentially help customer support, and particular advisors<sup>1</sup>, in both an active and proactive way, in their daily planning and to get a better overview of their work to solve issues faster. The overview tool will be referenced as both a tool and a page during this report.

## 1.2 Description of Telavox AB

Telavox AB started in the year of 2002 as a start-up by university friends, and has now grown to a company with around 200 employees, managing over 250 000 users. The head office is located in Malmö, Sweden and in total there are 21 different offices spread all over Scandinavia. Telavox' main business is called Flow, which is a platform for smart telephony, PBX, chat etc., and as of today they have more than 8 000 companies in Scandinavia using Telavox' platform

---

<sup>1</sup>An advisor is a dedicated customer support operator that handles issues for specific customers.



**Figure 1.1:** The dashboard in Partner as seen prior to the project. Blurred parts where sensitive information could be presented

Flow for their businesses. Their mission is to create a "wow-experience" when customers are using their products [1].

At the head office in Malmö there are around 40 developers working on Telavox' systems. Apart from them, there are also departments for sales, marketing, economy and customer support. An employee in one of their customer support groups is called advisor, and Telavox is aiming on getting each customer an assigned advisor.

An advisor is responsible for a set of specific customers by answering and solving the customers problems on demand. Today, the advisors are currently using the in-house developed web tool Partner to deal with customer requests and to solve problems that can occur both preemptively and when they have happened. Partner is an all-in-one CRM tool which contains all the functionality to solve customer problems that can occur. The goal for the advisor teams is to satisfy customer problems before they happen and therefore increase customer satisfaction. This is referred to as working proactively.

### 1.3 Thesis purpose, goals and questions

The CRM system Partner was originally built in the late 00's and has since then been exposed to several ad-hoc solutions. In Partner, at the first page displayed when logged in, there is an overview page called dashboard, with the purpose of facilitate for the advisor's daily work. Unfortunately, this dashboard does not serve its purpose since a lot of the presented information is out of date or irrelevant for the daily work of an advisor and does not help the advisor to get a good overview of the current workload or tasks at hand. The purpose of this project is to develop a new tool that makes the work for the advisors more efficient and to create a tool that enables an efficient work flow for the advisor.

The main focus is to analyze the old dashboard, in this report referred to as Dashboard 1.0, and develop a new dashboard that is called Dashboard 2.0.

The goals for the project are set up by the master's thesis students, the company Telavox and the institution at LTH. They are listed below:

- Learn more about interaction design.
- Create a product that is used in production
- Increase work efficiency on Telavox' customer support.

The goals together with the purpose lead to the following research questions:

- RQ1. What does literature say is a good process of developing a web tool with focus on interaction design?
- RQ2. How can work patterns and routines be analyzed to create a better tool?
- RQ3. How could an overview page within Telavox' existing CRM-system, Partner, be implemented?
- RQ4. How can an overview page improve work efficiency for customer service at Telavox and enhance Advisors understanding of what can be done?

These research questions will be answered throughout the report, and summed up in section 7.1.

### 1.4 Delimitations

The following delimitations are set up to avoid having a too large scope and getting an ever expanding project:

- The project is to develop existing tools and new ones in the "dashboard" specifically.
- The tool is to be focused only on the user group advisor.

- There is no possibility to satisfy every user requirement, but to make a general solution that satisfies and solves problems for our target audience.
- The project will not create a process for developing a tool, it is instead focusing on the product itself.
- The tool shall not replace existing tools other than the current dashboard in Partner.

The project is also developed with how the advisors work in mind. This means that implementation of mobile features and features for smaller screen sizes are not taken into consideration due to the way Partner is used, which is normally done in full screen on screens not smaller than 1920 pixels in width and 1080 pixels in height.

One thing to note in this project is that the header of Partner was not changed. The focus of the design was put on the main dashboard functionality. To delimit the project down to this part of the site and not to solve sole navigation and search questions helped focus the efforts to specific features on the site.

## 1.5 Stakeholders

The project has several stakeholders and they are presented in this section. Some of the stakeholders were known in the start of the project, but some came to light during the project. The stakeholders in this project are:

### **The thesis workers**

The project's main stakeholders are the thesis workers. The main work is done by these stakeholders and has the main responsibility in every aspect of the thesis.

### **Faculty of Engineering LTH at Lund University (LTH)**

LTH is a stakeholder that views the project from an academic standpoint. This aspect is of great importance due to the thesis being academic and the base for the project.

### **Telavox advisors**

The projects goal is to help the advisors in their daily work. This makes the advisors the central external stakeholder in the project. They are the subjects for the interviews, observations and general meetings during the project.

### **Telavox advisor management**

Team leaders in the Telavox advisor organisation has a need for more overview and information to analyze.

### **Telavox development department**

Due to the project being developed on a product with ongoing development

and several developers working with the code potentially years in the future, everything that is published is of relevance to the development department.

## 1.6 Previous experience

Carl and Johan are from the electrical engineering program with experience in programming in several languages. Johan has also previous experience at Telavox, where he did a two months internship and spent his time being a developer in the Partner team. In agile methodologies, Johan has experience in both XP and Scrum. Carl has no previous experience at Telavox, but has previously worked at Ericsson as a developer.

## 1.7 Report structure

This report follows the same structure as the implementation of the thesis, which contained literature studies (theory), methodology, an investigation of current situation, design and implementation, final evaluation and lastly conclusion and discussion. More information about these phases and its purposes can be read in chapter 3. The phases were mainly planned to be done in a linear fashion, but overlapped in some parts of the project. The whole report process can be seen in figure 1.2, and where it is applicable the chapter numbers are marked before the name of each phase, i.e. writings about the phase Methodology can be found in chapter three of the report.

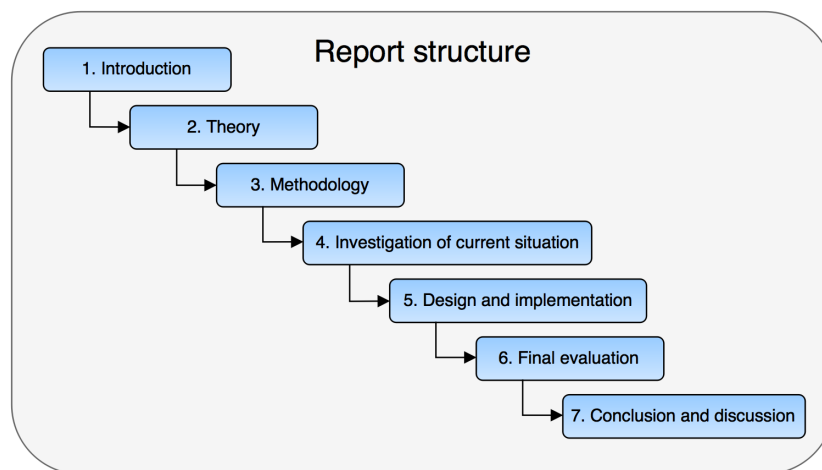
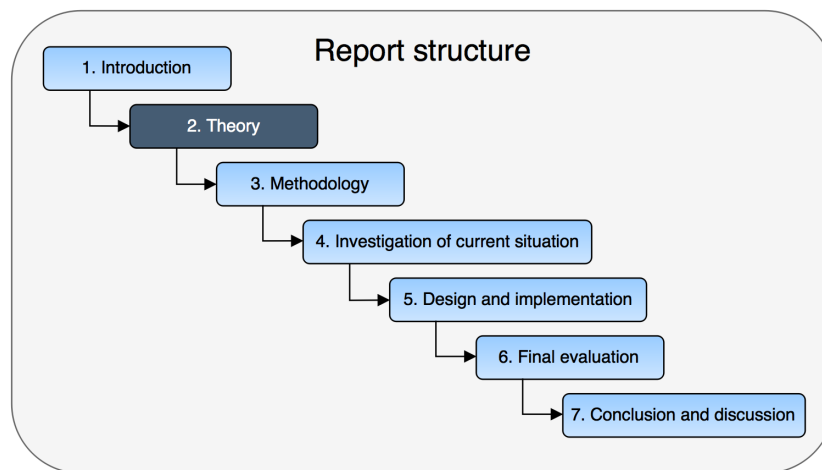


Figure 1.2: The project overview.

In this chapter of the report theory that is grounded in academic literature such as papers, books and websites are presented. The goal of this part is to add a foundation to stand on while reading and working on the master's thesis, and to be able to motivate decisions that can be of importance during the project. The chapter's purpose is also to answer RQ1 (see section 1.3). The part of the report process is highlighted in figure 2.1.

The first part of the theory section is focused on investigative methods for the current situation, where theory about several data collection methods are explained. The second section explains design theory and things that can be of interest when designing on the web. Further, a section on development explaining the theories of software development and integration. Lastly, a section on usability testing is presented.



**Figure 2.1:** The report structure with current chapter highlighted.



## 2.1 Data collection methods

### 2.1.1 Data collection methods introduction

To draw conclusions and analyze results in the project, data is needed to be able to determine how things are perceived and what can be improved. To get this information, data collection in various forms are done. Data can be categorized in to two main classes: quantitative- and qualitative data.

**Quantitative data** is data that can be generalized. It is often collected from large pools of data to ensure that it has a statistical quality. It relies on the notion of data being measurable [2]. A quantitative analysis is defined as connecting values to attributes that describe a phenomenon. This can be done in several ways, but to get the most out of it, it is important to have the purpose of the analysis clear. A good structured process for the quantitative analysis is the GQM-method (Goal/Question/Metric), which includes three phases [3]:

1. *Definition of the measurement*: Defining the goals, questions and measurements.
2. *Data collection*: The collection of the data.
3. *Interpretation of data*: Analysis of the data and answering of the questions in the first phase.

**Qualitative data** on the other hand is based on subjective feelings and thoughts. This data does often come from a person that is interviewed or is asked to describe how things feel or are perceived. The data is collected from smaller samples and can be a great way to use data to discover new ideas [2].

### 2.1.2 Observation

To collect information and get understanding of a system or get information on how users use certain tools, a system observation can be done. Observations are done by observing things that occur in certain situations. An observer can have different levels of interaction, from not interacting at all to being an active observer by asking questions or even participating [3].

In observations where the observer is noticeable, as sitting besides the observer for example, there may be an impact that affect the phenomenon studied and needs to be taken in to account [3].

### 2.1.3 Interviews

To elicit information from users about what they need and feel or get new information, interviews can be a valid approach. There are generally three ways

to gather information with interviews that lead to different results that can be analyzed in various ways. The ways an interview can be structured are open, partly structured or structured [2][3].

Using an open or aimed open interview is a way to gather qualitative information to reflect the interviewed subjects view on the interviewee at hand. The goal is to contribute new information, to enlighten parts on the topic that is not given [2]. The goal is not to give a general conclusion about the topic in an open interview [2].

The open interview has general questions at hand that is asked to the interviewee, where the person is free to talk about how the interviewee feels and interprets the subject. By aiming the interview to specific areas of interests, one can get a picture of individuals' interpretation of the narrowed topic. To not influence the responses, open questions are important and trying to repeat and get the interviewee to explain in greater detail what the interviewee just said [2][4].

During a qualitative interview it is important to get nuanced responses filled with subjective interpretations [2][4]. Else the result can fall flat and not give anything to the people analyzing the result of the interview [2]. To do this, questions are to be constructed in a way that they are open and to get the participant to explain their thought process in more detail [4].

An interview plan is a written plan for the structure of the interview. The interview plan is a good way to prepare for interviews and can help increase the quality of the interview itself [2]. It has aspects of the interview structure, what questions to ask, what subjects to move on to, introduction and ending. This is important to get a structure on the interview and to try asking one question at a time as not to get conflicting or confusing answers. In an aimed open interview the interview plan has questions regarding different subjects that are of interest with general followup questions that might be asked. There is no need to state precise questions in the interview plan, as the interview might take a different direction then predicted [2].

Interviews are usually documented by recording them, and notes can also be taken during the interviews to note the direct observations and feelings during the interviews [3].

#### 2.1.4 Analysis

After the gathering of data, the next step is to analyze it. The interviews and observations of the qualitative data collection might have resulted in a lot of raw data, but without a proper analysis and concretization it will not create any utility. The purpose of the analysis is to categorize and find connections between the data to find support for the coming design. There are several methods for analyzing data, and in this section one successful work flow is described [4].

**Affinity diagram** is a method for organizing ideas and data, which involves collecting facts from the notes and trying to connect them with each other. In short the process is done in four steps [4]:

1. Each person in the group writes down the most important observations on post-its.
2. Organize the post-its after similarity or how they relate to each other.
3. Discuss what is distinct with each group of post-its and name it with a short phrase.
4. The last step involves drawing lines between groups and themes, which will result in a hierarchy describing the relations between observations and ideas.

**A user persona** is a fictional person that can be a user of the system that is being designed. It is a tool to help designers solve questions about the design. It takes the abstract notion of "users", where every designer has its own view of who they are, and creates a concrete person designers can base questions of [4].

Based on information gathered from users, interviews and other information, one can create a user persona. The goal is to create a character with a name, picture and personality to make the persona come to life [4].

In a project there can be several user personas. Even if there are several personas there is always a primary persona that the design is created for. Secondary personas can still contribute to the design, but only if the goals of a secondary persona does not go against the main goals from the primary persona. The different personas should be done, such as that they are distinct and can not be confused with one another [4].

Creating a user persona is done in several steps:

1. Gather information about the roles that exist and divide them in to separate groups based on the interviews.
2. Get familiar with the data that is relevant for each different role.
3. Behavioral variables identified from an affinity diagram analysis of the interviews are then identified.
4. The different individuals are then grouped with these variables and a user persona starts to emerge.
5. Goals and distinctions are made for the different categories, and if several personas emerge they shall be made distinct.
6. Lastly, the user persona gets a background, complete with a photo and some back story. This is to create understanding and empathy for the user persona.

[4]

**The analysis of quantitative data** is done by using methods based on statistical analysis. It can be done to either confirm a theory and strengthen already known data, or it can be done to get an understanding of the subject that is observed [3].

## 2.2 Design

### 2.2.1 Design essentials

Norman describes in the book, "The Design of Everyday Things" [5], some necessary concepts of design. They lay down the basic ground for human centered design (HCD) and how humans interpret and have relationships with objects and other design solutions. The psychological concepts are based on how humans interact with and discover how to use an object. These concepts are the following: *affordances*, *signifiers*, *constraints*, *mapping*, *feedback* and *conceptual model* [5].

**Affordances** are what the object characteristics are to the individual user. They can be described as the relationship with the user from previous interactions with the object or similar objects.. For example, the average human has the knowledge that glass is transparent, but also that objects cannot move through glass (with exception for the quantum particle level, where other rules apply). The second example is an affordance that may be hard to discover. This can result in various accidents, such as humans or animals failing to notice the glass and walk in or through it [5].

**Signifiers** are ways to communicate an object's affordance. They put out information about an object's characteristics in various ways. There exists both intentional and unintentional, and even accidental signifiers. They can be deceiving in bad design, where some designs can indicate characteristics that are not true to the affordance [5].

Combined affordances and signifiers are all around us in our daily life. If a person notices that it is wet outside on the pavement, the conclusion the person draws is that it either rains outside or that it has rained recently. In this example the (unintentional) signifier is the wet pavement, whereas the relationship from previous experiences the wet pavement indicates the weather's current or previous behaviour [5].

**Constraints** are boundaries that tell what operations are possible. There are several types of constraints: physical, cultural, semantic and logical. Physical constraints are based on properties of the real world and tell the possible actions one can do with for instance an object. Cultural constraints are based on actions that is allowed in certain situations based on the individual's values and

experiences from the culture one is from. Semantics is based on assessing the meaning of a situation or context, and the constraints comes from the possible outcome based on these. Lastly the logical constraints are constraints that can be logically derived [5].

**Mapping** is the bound between two entities of an object and how they are presented. Taking for instance an object with a button array that each turns on one of two lamps on each side. The object has a mapping to show which lamp the button turns on. Hopefully in the array, the right button turns on the right lamp, while the left lamp is turned on by the left button, thus using a mapping correlating to how the user sees the lamp [5].

**Feedback** is an important design element that lets a user know when an action has occurred. There are several ways of delivering feedback information such as visual feedback or feedback via sound. As important feedback is to the user, where lack of feedback can be very bad, poorly designed feedback, such as too much or too excessive, can also be bad [5].

**Conceptual model** is the user's model that is their own built version of how something works. This is an interpretation of how the object is designed and works. The conceptual model can be developed by reading manuals, by another user telling them their model or experience. Conceptual models are often not a complete model of the system, and they can even be wrong. The important thing for the design is to convey a conceptual model that is useful [5].

### 2.2.2 The three design phases

While designing, especially in the starting phases, the uncertainty of what is going to be made and how it will look can be very high. Problems like not knowing what to implement and if it fulfills requirements, to how it effects the user of the system in a positive or negative way, are usual. To make the design process well defined and to get a clear view of what has to be done in the design phase, there are certain procedures you can take. One of these procedures is to split the design project in to three phases [4].

These phases consist of *the conceptual design phase*, *the processing phase* and *the detailing phase*. Each of these phases contribute to the process of creating an interaction design, but do not have distinct boundaries, which make the process nonsequential [4].

#### Phase one: Conceptual design phase

To create a product that has value and is fit for users, it is important to understand who the users are and how they use things similar to what is to be developed. In short, to get this understanding the best way is to simply ask the users and write down what have been heard and seen. This is the first part of

the conceptual phase, which is all about finding out what is going to be built. This includes creating a vision, goals, and exploring different conceptual ideas [4].

In general, the phase includes two activities, sketching and finding out needs, and shifting between them. It is possible to either start sketching or asking around about the use cases and finding out needs as long as the shifting is not forgotten.

### **Phase two: Processing phase**

The conceptual design phase will result in knowledge about what to develop and why, but not really how. The Processing phase involves processing the information from the first phase to conclude what functionality and content the design will contain. Moreover, the phase involves determination of how the user interface should look like. During the processing it is critical to make several variations of the sketches and always rethink and come up with other options. If not, the group can get stuck on one idea and design that may not be optimal in the end, and keep working on something that does not lead anywhere [4].

The processing phase is also about stating requirements. The requirements play several roles during the development process. They can for example ensure the customer what they will get in the end, but also be of support for the developers during the implementation [6].

When requirements and functionality has been decided, the next natural step is to start sketching the layout and interface. In general, sketching can be done in two ways, low-fidelity and high-fidelity prototypes. In this phase the focus is on the low-fidelity prototyping as the high-fidelity is covered in the next phase.

*Low-fidelity prototypes* are according to Sefelin et al. [7] defined as visualization of design ideas at a very early stage of the design process. A vital thing about low-fidelity prototypes is that they are simple enough to create an understanding of the design, but are also not time consuming to develop. A common way to create low-fidelity prototypes is to sketch with pen and paper.

However, the low-fidelity prototypes can also be visualized on a computer, and are even preferred by the test subjects in most cases, and lead to the same user statements as paper. On the other hand, it is still possible that an early low-fidelity prototype will make the user focus on wrong things, like graphical details, or even keep quiet about criticism, because the design group already has come so far in the process in the eyes of the one observing the sketches [7]. As an extension to the low-fidelity prototype it is possible to put several sketches in a flow, which can then simulate the product in an early stage with different views [4].

### Phase three: Detailing phase

In the last phase of the process of the three phases, a *high-fidelity prototype* should be developed. A high-fidelity prototype is a realistic simulation of the product, including graphics and interactions. This is more time consuming and less changeable, but might ease the process of giving the users an overall impression [8]. A high-fidelity prototype can either be developed as just a prototype and then discarded, or developed as an evolutionary prototype. An evolutionary prototype is an early implementation of the final product which can be developed iteratively [4]. Arvola claims that an evolutionary prototype takes at least four times as long time to develop as a temporary. To reduce time spent on prototypes a good way is to hard code variables or use some kind of presentation software such as Apple Keynote or Microsoft PowerPoint.

#### 2.2.3 User in focus

Designing tools and software that are intended to be used by people should have focus on the user experience and usability. Traits, such as easy to learn, easy to use, useful for the task at hand and have functionality that is needed for the user to do their work, are of importance to take into consideration [9]. Evaluating that the product being developed has these traits can be hard for a designer or developer, and therefore it is important to understand the user and take advantage of the knowledge and expertise they possess.

To get an understanding of the user early can be very beneficial due to input in various problems that can be of no mind to the designer or so obvious to a person invested in the project that it has been overlooked. Important is to understand the users instead of just identifying and stereotyping them [9].

Prior starting the design of a system, a process to understand and involve these users should be started. This can reduce costly mistakes that can consume both time and money to revert when the project is well under way [10]. Using the users perspective and influences early can make the design decisions more easily verifiable and reviewable early. This can open new design perspectives that previously might not have even been thought of. In other words this can be described as working proactively [9].

There are recommendations to involve the user as much as being a part of the design team, or letting them sign off on the design before developing it. Other methods involve letting the user participate and be able to agree on the design, but ultimately having the design team taking in factors as cognitive and emotional factors as well to make the final decision [9].

However, it is important that the designer is critical to the client's opinions because they often do not know exactly what they need. Even if it is clear for both the client and design group what will be developed, it is not recommended

to skip the conceptual design phase, mentioned later in section 2.2.2, since it is the design group's responsibility to explore other possible options [4].

### Don't make the user think

According to Krug [11], the most important thing when it comes the usability of an application is: "Don't make me think!". An application or website should be self-evident or at least self-explanatory in a way that the user does not has to think about its next move. An example is whether an object is clickable or not. In figure 2.2 three links are shown with different formatting, which makes the user think differently. In the left example it is obvious, the link is clickable since it is generally accepted that a button is clickable. On the other hand, it would be hard to distinguish the link from the text in the right hand example. Instead of creating a distraction for the user and add to its brain workload on whether a link is clickable or not, the better way would be to just make it self-evident. Krug mentions, one solution for this can be to use conventions or well-known frameworks.



**Figure 2.2:** An example of clickability and the use of well-known framework, in this instance Bootstrap [12].

Further, the users seem to prefer mindless choices. Krug's second law of usability says:

*"It doesn't matter how many times I have to click, as long as each click is a mindless, unambiguous choice."* [11]

On the other hand, many web designers and usability professionals have spent a lot of time debating how many clicks, at maximum, a user must perform to be able to reach a specific target on a website. Krug argues that a single click that requires thought equals three mindless and unambiguous clicks. In that case the number of clicks are irrelevant for the user and the page can therefore be designed with main focus on the Don't-let-the-user-think-principle. This theory has been confirmed by research consultant Joshua Porter in 2003 where he studied 44 users attempting over 600 tasks. Porter draws the conclusion that the satisfaction of users does not depend on the number of clicks, but of the success of finding what they are looking for [13].

The human is lazy in its nature. When a user opens a web page the user tends to spend very little time reading. A common problem when creating websites is that developers designs the site assuming the user will read the whole active page, to continue reflecting, determine what is next and at last click on the most reasonable link. This is not the case. What the user actually does is scanning the



site for content that are either interesting or what the user actually are looking for and clicks on the first choice that seems logical. Yet again, it is important to use design conventions and eliminate distractions, but also break pages up into clearly defined areas [11].

Furthermore, a user does not think about how things work, most people just muddle through. A common example is that people does not really know what they are doing, but are doing it anyway as long as it is working. Very few people take the time to read instructions and are instead trying to get things done by trial and error. This depends on the fact that people do not care enough and that if a user finds something that works, the user will stick to it. This can result in a potential problem where all the users are using an application the unintentional way and can therefore decrease efficiency in the long run. Because of this, it is important that a page is always designed in a way where the intentional flow of a use case, also is the logical way to do it [11].

### **Visual noise**

The noise on a web page is defined as everything that create a distraction for the user in a way that prevents the user from getting the actual message of the product. Too much content or disorganized content are two usual examples which creates visual noise. A good principle when designing a page is to assume that everything is visual noise until it is determined that it is actually contributing to the page [11]. According to Mackworth [14], visual noise causes tunnel vision, which probably is the underlying problem when browsing at site with too much content. It makes the user focus on the wrong content of the site.

## **2.3 Software development**

### **2.3.1 Software development introduction**

In general, when developing software there are two methods of working. These two are Waterfall and Agile. The main difference is that Waterfall has a linear structure whereas Agile has several different parts of the project ongoing in parallel. Development at Telavox is usually done in an agile way, using various agile methodologies in different teams. This section will describe agile development in general.

### **2.3.2 Agile in general**

From the Agile Manifesto website [15] four general main guidelines for all agile methodologies are stated:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiations*
- *Responding to change over following a plan*

Worth noting is that the manifesto also states that there is value in the later things mentioned in each guideline, but the first mentioned items are usually of greater value.

The goal of agile development is to welcome change and to continuously deliver software that is of value to the customer. The aim is to create a system that is simple enough and has the agility to be able to change. It also highlights the importance of communication between people, both within the project with internal communication between business and development, but also between the customer and developers [15].

To reach the goal of continuously releasing a product and having high agility the agile work procedure is done in quick iterations (usually weeks), combining many usual project stages done separately in a waterfall model to be done simultaneously.

Almost all agile methodologies are done with features broken down in to smaller tasks that are assessed in some kind of way in regards to the estimated time it takes to get done. This is to estimate what can be done in a fixed period of time and to be able to fill this time with various tasks.

### 2.3.3 eXtreme programming

Extreme programming (XP) is an agile methodology and implements test-driven development as well as working in pairs through development. As with compliance with the agile work method it handles several stages of development at once, and iterates through phases [16].

Due to pair programming, the code that is being written is reviewed during the development. The "driver", the developer writing code, gets his code reviewed and questioned by the "passenger". The review process is done while the coding is done, and has the potential to improve the code during development.

Heavy emphasis on test-first, high delivery rate, refactoring and pair programming are values that are of importance in XP [17].

### 2.3.4 Company development and integration

As Telavox is a highly agile company, new features are developed parallel with each other. To facilitate those needs, development at Telavox is done in the

version control system Git, and distributed by the service Github. Using these tools together with a branching strategy known as "Branching by task" [18], developers create a branch from the main line of code called "master", make their change or implementation, then connects the code back (merge) with the master branch. This allows several developers to avoid problems like the shared data problem, the simultaneous update problem and the double maintenance problem, which is essential to solve in the world of software configuration management [19].

As a developer works on a specific task, the developer creates a branch from the most recent code in the master branch. This is then worked upon until the engineer considers the feature complete. The code is then pushed out to Github and a pull request is created. This request is reviewed by one or two other developers who share knowledge and they try to give comments about good and bad features in the code that is to be deployed. As the code gets accepted in the review, the pull request is merged in to the master branch and put in to production.

## 2.4 Usability testing

### 2.4.1 Usability testing introduction

In most projects, doing usability test is a valuable exercise to identify faults or weak spots in a design. By focusing on good usability and how to use the product, usability testing can help improve effectiveness and create less errors and a better user experience. It can also be seen as a tedious task to do since it requires planning and scheduled work time that usually takes days and involves high costs [11][20]. Several papers regarding cutting down both cost and scale on usability testing has been written [21]. Methods for implementing these methods of discount usability testing has been assessed with for example agile work methods [21].

This method of discounted usability testing has a heavy emphasis on being fast, easy and simple to implement. It also focuses on qualitative data that has a spotlight on being as descriptive as possible and analyzed by assessment, insight and sorting, instead of quantitative data that is analyzed statistically [3][11]. When analyzing the data the goal is to make usability and productivity improvements by interpreting the observations of the users [20][21].

As stated, the focus is on gathering insight and not proving facts, therefore the number of users participating in gathering information about the usability, called usability test, is usually limited to a low amount, around three to five people [11][22]. This makes the usability testing easier to conduct in a small time frame.

Several methods to extract data can be used. Two of them are described in the following section.

### **2.4.2 Scenarios and thinking aloud**

Scenarios use a version of the system that is able to simulate the user experience and let the user do tasks that are predetermined [21].

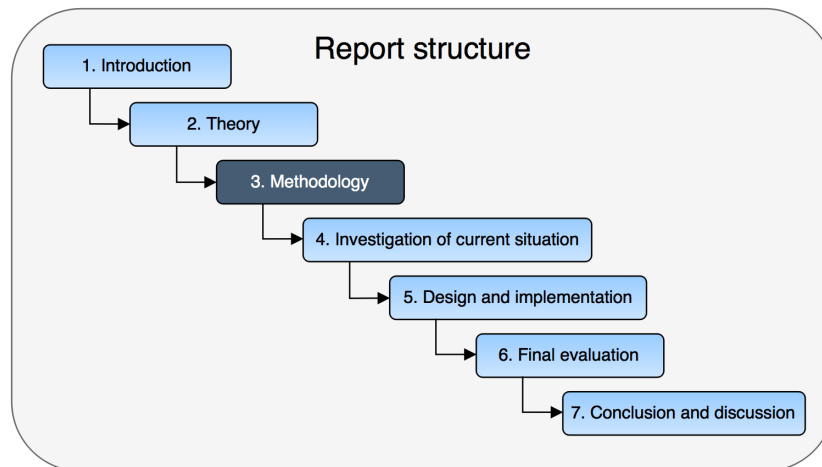
Thinking aloud is an interview technique that is based on letting a user do tasks with the system while the user is telling the interviewer what it is doing. The user is encouraged to describe actions and expectations of the system as it is performing tasks. The system can be prototypes of both low- and high-fidelity [21]

---

# Methodology

---

In this chapter the various methods in this thesis are motivated why they were presented and used in the thesis. In figure 3.1, one can see the part where this chapter belongs in the report structure.



**Figure 3.1:** The report structure with current chapter highlighted.

## 3.1 Documentation

During the project a daily journal was kept, describing the things that was done and discovered that day. This was to be able to see what had been done and to get an overview of how the work progressed. Moreover, a specific document was kept for the more comprehensive things that was accomplished. This document was often the first thing to fill in after e.g. an observation or interview. It detailed the purpose of the activity, result and other various things that might

be of interest in a later stage. This was also to get an overview and record what was done, but also to instantly reflect on the task that had been finished.

During the project, focus on writing the report continuously was done to ease the work load in later stages of the thesis, but also to increase and repeat knowledge learned and to avoid forgetting things along the way. By doing this, the report became more structured and that things falling through the cracks could be avoided. The report was written by both thesis workers, together as well as separately, where every part was reviewed by the other thesis worker. Where separate work was done, it was estimated to be equally divided.

## 3.2 Preparations

This phase consisted of two parts. **The first part** was the literature study that resulted in the theory part of the report. In this stage the main practical knowledge was gathered to create a base for the project and to get deeper domain knowledge. The gathering of literature was done by various methods. In the very start of the project, basic literature was provided or recommended by the LTH supervisor. Other ways of gathering literature were by using academic databases as IEEE Explore and Google Scholar with various key words as "*interaction design*", "*user experience*", "*user engineering*", or a combination. Also inspecting similar projects done at the Department of Design Sciences and observing their references helped finding relevant literature. This was done due to previous experience when writing reports with good results.

Theoretical sources were evaluated on the basis of origin and age, with some exceptions. However, a published book was still valued higher than a website, because of its publisher and meticulous examination [3]. Therefore, it was natural to look for basic information in well-known books within the field, to continue and look for the latest research in other published media. The used research papers were mainly from known researchers in the field to ensure their credibility.

**The second part** was the methodology stage, where based on the theory and knowledge accumulated in the literature studies several methods were chosen or developed and planned. This was to provide a rough overview of how the project was to be done and with what methods. The methodology chosen in this section was done by evaluating how Telavox worked and how the project could be completed in the best fashion based on previous experience to fulfill the research questions. Many methodologies was also chosen due to previous personal experience with them.

## 3.3 Design development

### 3.3.1 Design development introduction

The design development phase began with a collection of information (section 3.3.2) to be able create fundamental requirements and user personas. With this information it was then natural to begin the designing and implementation (section 3.3.3), where iteration 0 initiated the phase with several developed low-fidelity prototypes. These prototypes were evaluated to be able to change the design in an early stage. This was the end of iteration zero. The next two iterations were similar but instead of the low-fidelity prototype, an evolving high-fidelity prototype were developed. In the last part of the design development, a final evaluation (section 3.3.4) was done. The design development process in this report is covered by chapters 4-6.

### 3.3.2 Investigation of current situation

The investigation process can be seen in figure 3.2, where all the methods in the figure are described below.

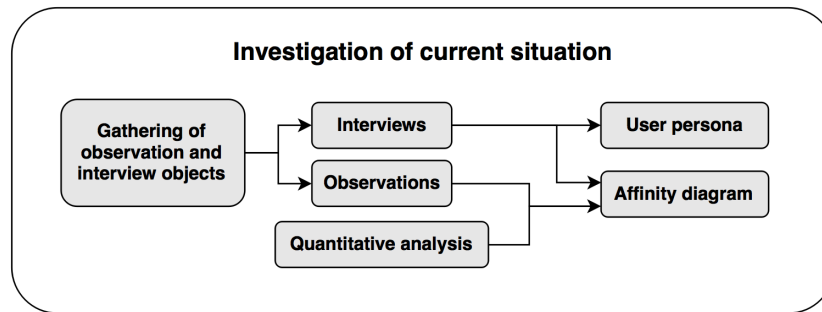


Figure 3.2: The process in the investigation phase.

**Quantitative analysis** was done by using Google Analytics. This method was chosen due to its ease of implementing in to Partner, and the huge amount of data it could give. The purpose of the analysis was to compare the quantitative results to the results of the observations and interviews to confirm our conclusions or show a different perspective. Because of this, the quantitative analysis was not the main source of information but to confirm observations and interpretations of the qualitative analysis.

**Gathering of observation and interview subjects** was done to prepare for the observations and interviews. This was not grounded in theory, instead it was made in a brainstorming session on how to reach out to all advisors and get interested people who could be interviewed.

**Observations** were used to get a deeper understanding of the system and to elicit more information about how advisors use Partner.

**Interviews** were conducted in an open, aimed way to collect qualitative data. This to get a feeling of how the advisors worked and who they are, and to collect subjective thoughts and feelings that could identify current work patterns, how they felt it was today, and to discover new things to implement and think about during the development.

**Affinity diagram** was chosen as a way to analyze and group the results from the interviews, but also from the observations and the quantitative analysis. This to get an overview of all data collected and to identify important categories that might not have been discovered or stood out prior to the affinity diagram. The affinity diagram resulted in requirements for Dashboard 2.0.

**User persona** was also a result from the interviews. It was done mainly to create a more clear and common picture of the user instead of having two divergent ideas of who the users of the system is.

### 3.3.3 Design and implementation

Implementation of the project was divided in to one starting phase, called *iteration 0* and two other implementation phases that was done iteratively, *iteration 1*, and *iteration 2*.

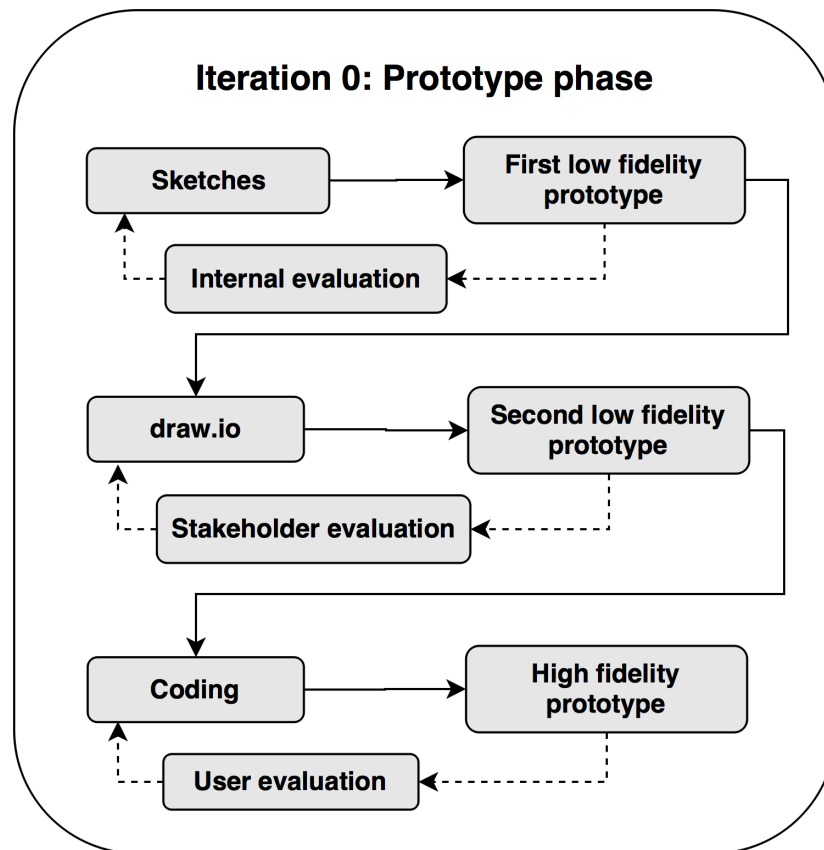
**Iteration 0** (figure 3.3) started with using all the knowledge gathered in the conceptual design phase, to create sketches so that an overview of the design could be created as stated in section 2.2.2.

Both participants in the project made sketches separately to not influence each other. The sketches where then shown in a meeting, discussed from a design perspective and the result was put together in to three sketches. Mock ups using *draw.io*, a powerful drawing tool used from Google Drive, were made and the sketches and mock ups were shown to stakeholders to verify that correct information was collected and interpret during the investigation phase. Together with stakeholders a final sketch was developed to be the design the project worked against.

When the high-fidelity prototype was created, a famous front-end framework was used, Bootstrap. It contains HTML- and CSS-based design templates for several interface components that is commonly used on internet, and therefore also well known. Developing front-end is often time consuming, but with Bootstrap it is possible to save a lot of time and at the same time design buttons, menus etc. in a way that an average user would recognize.

The browser that was used during development and also for testing in first hand was Google Chrome 51. This was because of Chrome's developing tools, which



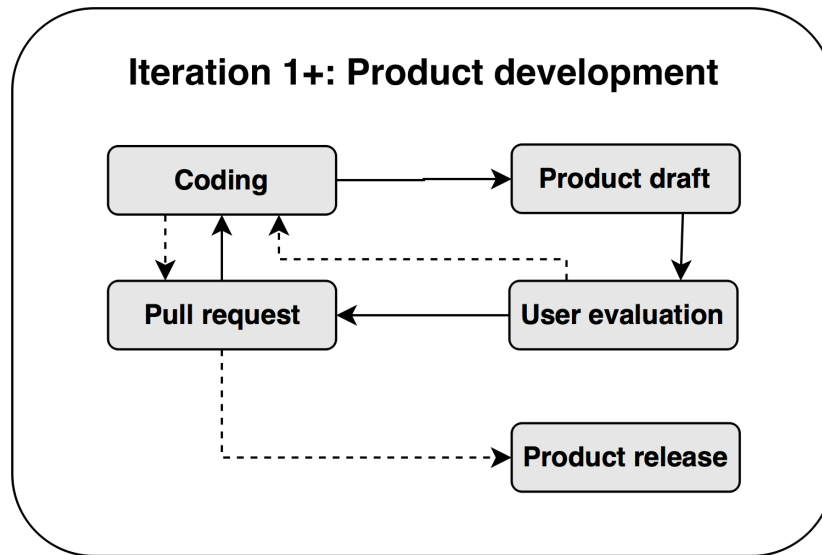


**Figure 3.3:** The process in iteration 0. The solid lines represent direct, required path, while the dashed lines represent an iteratively, optional path.

are great to use during development and troubleshooting. Besides that, our opinion is that Chrome has better compatibility, on the web, overall.

**The following iterations** after iteration 0, were in general outlined in the same way, starting with coding that resulted in a product draft, followed by user evaluation. Depending on the result of the evaluation, the development continued with either coding or a pull request. When the pull request was merged into master, the iteration ended with a product release. The process for iteration 1+ can be seen in figure 3.4.

Software development was done mainly in two coding languages: Java, JavaScript, with several libraries and technologies, including JSP, JSTL, jQuery, Stripes and AJAX. This was done to keep the code consistent to the existing code in Partner, but also to implement some more modern features such as AJAX due to its usability benefit of asynchronous HTTP-requests. At the same time the



**Figure 3.4:** The process in iteration 1+. The solid lines represent direct, required path, while the dashed lines represent an iteratively, optional path.

resulting code kept an industrial standard and our previous experience with it made it an evident choice.

During the development the coding were divided between Carl and Johan, which made it possible to work simultaneously in an agile way. Changes were constantly pushed to the Telavox' GitHub repository to make sure both had the latest update and furthermore avoid unnecessary conflicts. When there were doubts about any design decision, they were discussed instantly as development took place at Telavox' main office in Malmö.

Extreme programming was not implemented during the project, mainly due to the code review process at Telavox would make the "pair review" redundant. Some times pair programming was used, and a typical "driver" and "passenger" relation emerged. This was however not consistent through the project. As there were a lot of front end changes and changes, code testing was neglected.

### 3.3.4 Final evaluation

The evaluation of our result, Dashboard 2.0, was done in three different ways, which were chosen based on availability, ease of implementation and how it could help the project answer the proposed research questions.

To be able to compare the new Dashboard 2.0 with Dashboard 1.0, an analysis of the old design had to be done. This analysis was based on observations of

usage, the interviews and theory gathered during the literature study.

The second evaluation method was to involve the User Experience team at Telavox, who has expertise in the user experience area from a different view than the thesis workers, and could help find flaws or things to improve in the design. The evaluation was done to take advantage of the expertise the team could provide to the project.

The third method was to do user tests. These tests were the final way to evaluate Dashboard 2.0 and to get a view of the result of the project, the challenges that still exists and to see if the project has a positive impact on the advisor work flow.

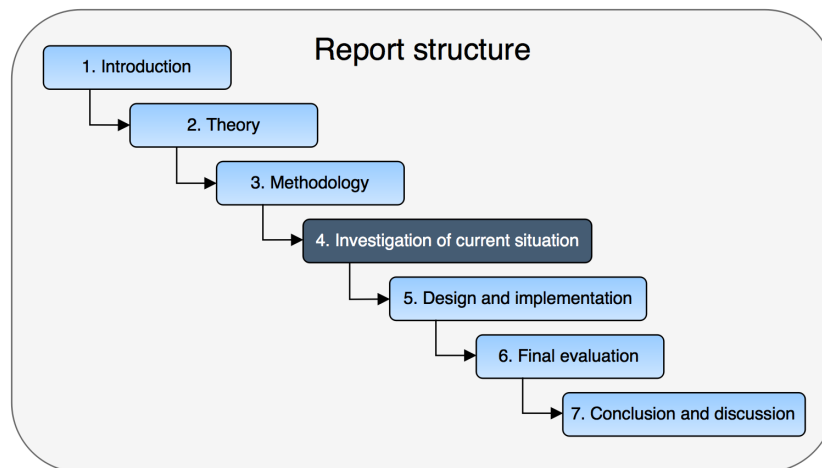
---

## Investigation of current situation

---

In this chapter the investigation of the current situation is described. The investigation is a phase, corresponding to the conceptional design phase mentioned in section 2.2.2, to understand the product, context and the users, but also behaviour of the current work flow. In this chapter, RQ2 (see section 1.3) will be answered.

This section consisted of both quantitative- and qualitative data collection. For the quantitative data collection Google Analytics was implemented, and in the quantitative case observations and interviews were conducted. How they were implemented and results are presented in this chapter together with other methods and results. In figure 4.1 one can see the part where this chapter belongs in the report structure.



**Figure 4.1:** The report structure with current chapter highlighted.

## 4.1 Quantitative analysis

### 4.1.1 Application

The first thing in the investigation phase was to do an implementation of a tracking system in Partner. Some research was done on Google Analytics and how it could be used for analyzing Partner. The implementation of the analysis was done over a five day period during a work week, and followed the GQM-method with following definition:

*The purpose of this study is to analyze the dashboard,  
to determining how it is used today,  
by evaluating page views,  
with the point of view of advisors,  
in the context of Telavox.*

The main questions that were answered were:

1. How many advisors are currently using the dashboard continuously?
2. Which utilities on the dashboard are currently being used?

The plan was to get the first question answered by tracking how many times the dashboard was loaded directly after the advisor had been in a customer's page. This was not a perfect way to do the measurement but would give a hint of the answer.

The second question was answered by tracking which pages was visited directly after an advisor had visited the dashboard page. Since only the URL was recorded for the page views Google Analytics did not know that two different customer overviews actually was the same page but with different information on it. Compare following URLs.

`https://partner.telavox.se/.../overview.jsp?id=11111`

`https://partner.telavox.se/.../overview.jsp?id=22222`

This was solved by creating groups in Google Analytics, which grouped all the overview pages together.

During the implementation the code was tested on a local branch of Partner and with a personal Google account. A problem that appeared during the testing was that the statistics in Google Analytics was only refreshed once per day. This made the implementation protracted since the changes in the code, did not give results until the day after. In total the implementation took four days and two extra days to get the code deployed with review and upgrade of the system.

To be able to track a specific user some modifications to the default Google Analytics tracking code was done. Normally, Google Analytics will only track

a specific user according to a made up tracking ID. In this case there is no possibility to see which users that were an advisor and not, since the tracking code was deployed on the production version of Partner. However, in this quantitative analysis the purpose was to investigate only the advisors. This imposed some modifications of Google Analytics implementation, thus the Track ID functionality was implemented. Since Google Analytics had restrictions in the user agreement of Track ID, the internal Telavox Partner user id number was used as the identification. In this way it was easy for an employee with access to Telavox' user database to associate a user id to a physical person, but for everyone else the number was useless. Moreover, one extra custom variable (also known as custom dimension) was added to Google Analytics. This variable was set to the user id as well and was used to get access to a specific filter function in Google Analytics, which was not available with the regular Track ID feature.

#### 4.1.2 Results

The results of the quantitative analysis ended up in three different tables where several conclusions could be made.

Table 4.1 shows that 45.6 percent of all page views during a week came from an advisor. Furthermore, it shows that an advisor on average loaded the dashboard 5.42 percent of all loaded pages in Partner.

**Table 4.1:** Number of pageviews for all users and the user category Advisors during a week.

<i>Page</i>	<i>User group</i>	<i>Page views</i>
All Partner pages	All users (539)	56 963
All Partner pages	Advisors (32)	25 975
/dashboard.jsp*	Advisors (32)	1 409

Table 4.2 shows the page view that was loaded next after the dashboard. In the table it was easy to distinguish which functions that were used in the current dashboard in the test period. To make it clear which functionality that corresponded to which URL path, each path was paired with a functionality. The paths were picked with two criteria in mind:

- The number of page views to identify where most of the users went after the dashboard.
- All the current functionality on the dashboard were covered, to see which functions that were used and not.

With 48.57 percent of all the clicks it was clear that the ticket functionality was the most used and critical function. Further, the advisors were also often

trying to find a customer, either by the search function or by clicking on the customer directly in the ticket entry. This action was done 40.05% of the times. Documents, quotations and portings were reached 2.11 percent, 1.50 percent and 1.30 percent of the times respectively. Commission page was not reached at all.

**Table 4.2:** Number of views for each page in Partner accessed **next after** the dashboard. The asterisk (\*) is a wildcard character used to represent a number of characters or an empty string.

<i>Path</i>	<i>Page views</i>	<i>Percent</i>	<i>Used function</i>
/customer/mailticket.jsp*	373	30.30%	Tickets
/searchcustomer.jsp*	260	21.12%	Find customer
/messages/mailtickets.jsp*	214	17.38%	Tickets
/customer/overview.jsp*	127	10.32%	Find customer
/customer/account.jsp*	106	8.61%	Find customer
Other	85	6.90%	
*documents*	26	2.11%	Documents
/portings*	16	1.30%	Portings
/partner2/messages.jsp*	11	0.89%	Tickets
/ncg/ncg_start.jsp*	10	0.81%	Quotations
/crm/crm_start.jsp*	3	0.24%	Quotations
*bonus*	0	0.00%	Commision page
<b>Total</b>	1231	100.00%	

## 4.2 Elicitation of subjects for observations and interviews

### 4.2.1 Application

Simultaneously as the Google Analytics implementation the need for subjects to participate in observations and interviews arose. To get a hold of participants a survey was created through Google Forms, a free online survey software. This form was then sent to all advisors via the Telavox advisor mail list.

The email sent to the employees was written in a short manner that explained the premise of the whole project and that any feedback was appreciated. In the email there was a link to the form, where three short questions had been formulated.

The questions were in Swedish, translated to the following:

- Your name
- Do you use the dashboard?
- What is good and/or bad with the dashboard, suggestions for improvement?

### 4.2.2 Results

Of the thirty advisors the survey was sent to, ten answered. The expected answers were three to four, which made ten well beyond expectation and a surprise.

Unfortunately, the last question was actually three different questions in one. This gave an incoherent response where the answer could not be distinguished on what question they were answering. Instead the questions should have been asked separately to get the survey answers more concise and therefore also more usable. However, there was still the possibility to sort out the answers on later interviews. On the whole the main purpose of the survey was fulfilled.

The elicitation was mainly based on the survey. The answers were analyzed individually based on user engagement and what they seemed to prioritize, followed by a discussion where each of the thesis workers motivated who and why an advisor should be observed and interviewed. The discussion showed that each of the thesis workers had picked four respectively three candidates, where three of them were overlapping. In the end, the advisors were elicited based on following:

- Advisors who had opinions, and more specific opinions about interaction design.
- At least one advisor that was using the current dashboard frequently and one that did not.

For the observation our top two advisors were chosen and for the interviews all four were selected. However, during the interviewing period several advisors had vacation, which made us reduce the amount of interviews to three and change one of the advisors. In the end, the interviewees were three advisors from three different segments and positions.

## 4.3 Observations

### 4.3.1 Application

Trying to get a basic understanding of the domain, two observations were done. The elicitation of the observation subjects was done as described in section 4.2.



The observation was done on two individuals where one project member sat besides the advisor and observed the work flow during an hour, taking notes of what happened. A dialog was initiated when questions arose to get as big of an understanding of the situation.

### 4.3.2 Results

The result of the observation resulted in the following:

**A new role emerged:** team leader. This person was partly an advisor, but also had the administrative role of team leader of the team the person was a member of. This role included administering customers to advisors and get an overview of workload.

**The amount of customers** an advisor worked with. Depending on various advisor teams, these ranged from companies with one to ten users (where the advisor had upwards to 300 companies to handle), companies with eleven to 25 users (around 150 companies per advisor), 26 to 75 users (around 50 companies per advisor), to the largest of companies with over 75 users (a lower amount, maybe 3-10 companies per advisor).

**The use of the system.** How an advisor handled the system when a call came in. It differed in the two cases, but both used the internal ticket system to get an overview of what was going on, searched for customers in the header of the page constantly, handled and got an overview through the customer page and used various other functions during their sessions with customers. External services as Google Calendar and Google Drive were also frequently used during the session, where the work flow was interrupted to use these services.

**The wish for a better overview.** The subjects expressed the wish to get a better overview over their customers, were it was the amount of calls during a period of time or just quicker access to the customer profiles.

The results of the observations were documented and reviewed together to discuss what was learned. The results gave a foundation for the interviews and paved way to make the interview plan.

## 4.4 Interviews

### 4.4.1 Application

The interviews were conducted in one of Telavox' interview rooms. The structure of the interview was planned beforehand in an interview plan document, that can be viewed in appendix A, outlining questions and general topics that was going to be discussed during the meeting.

The interview plan was made as described in section 2.1.3. It had two parts, an introduction and a main question part. The introduction outlined the purpose of the interview, the amount of time it would consume, the fact that it was being recorded and the structure of the interview. The question part was divided into three areas: *about the interview subject, the subject's work flow*, and lastly *specific questions about the current dashboard*.

Questions for all these parts were made in an aimed open way to ensure that the answers did not go to far away from the subject, but still be of a subjective manner from the person interviewed.

Prior the interview an audio recording device and a laptop with a screen recorder with Partner was set up. This was to capture everything that was said during the interview and to be able to go back and watch various demonstrations that might have been performed during the interview.

The interviewers were divided in to two roles: the interviewer and the secretary. The roles were set to be fixed as every interview gave experience and would build to better results due to practice. In this project Carl acted as the interviewer and Johan the secretary. The roles had two main differences, the interviewer had the responsibility to drive the interview forward, while the secretary analysed the answers and came with various side questions if there was a need for it.

After every interview the results and questions were analyzed to see if there could be made any improvements due to the qualitative nature of the interview structure.

#### 4.4.2 Results

The interviews resulted in three 45 minute sessions with three advisors with various roles. The roles ranged from a regular advisor, team leader advisor, to head of the advisor department. The range of work experience at Telavox also differed, going from six months up to five years.

The common results from the interviews were:

- The general feeling was that there is some parts of Partner that is very slow.
- Partner can be big and complex at times.

These results were central concepts that were taken into account when developing the new dashboard. As one of the interviewees said: *"If the tool is slow or hard to use, people will not use it"*.

The goal of a dashboard, when asked, was to create an overview. All three advisors said that the dashboard today gave information about current active and awaiting tickets, but not as good an overview that was desired. The feeling was that improving the overview of an advisor's and team members' tickets could

increase the work efficiency. The ticket overview function was very central in all interviews, and also noticed as an important part in the observations.

At the various levels of roles, the demand for different functionality started to appear. Advisors prioritized information about incoming tickets and to solve problems more in a proactive way with the help of tools. Team leaders, on the contrary, wanted to know the workload of their team, have information of incoming and outgoing calls and so forth.

The need to work proactively was regarded as something that was important, and features to help advisors work in this way were commonly suggested. To create tags that registered what type of common problems customer has, group them and analyze was one suggestion.

The result of the interviews was further analyzed in the affinity diagram (see section 4.5.2).

## 4.5 Conclusions

### 4.5.1 Evaluation of quantitative data

The quantitative analysis was summarized in six conclusions:

- Almost half of all the actions in Partner was done by an advisor.
- The dashboard was being used continuously by the advisors.
- The ticket functionality on the dashboard was widely used by the advisors.
- Advisors needed to find a customer often.
- Portings, documents and quotations were seldom used.
- Commission page were never used.

Since the advisors was in total 32 persons, compared to all users in Partner, the quantitative analysis confirmed the importance of the stakeholder group advisors in Partner and also the delimitations of the project.

According to the results of the analysis the ticket functionality was the most important, which did correlate to the interviews and was therefore kept as a priority. The search functionality was not directly associated to the dashboard, which made it irrelevant for the project. Portings, documents and quotations were not specially mentioned as important in the interviews, and at the same time seldom used, which made these three low priority. Commission page was ignored since it was never used.

### 4.5.2 Affinity diagram

The affinity diagram was mainly created from the output of the interviews, but also the observations and the quantitative analysis. The process was done in three steps. Firstly, Carl and Johan worked individually, where the important conclusions from the interviews were identified and then written on separate post-its. These post were afterward sorted and categorized in groups of similarity. The categories were the same as the three question areas, that were used in the implementation of the interviews: *about the advisor*, *Partner* and lastly *the dashboard*. See section 4.4.1 for more details.

Next, the groups were analyzed individually to see if there was any possibility to create smaller sub categories. Within the dashboard category, the sub categories used were *goals*, *features to keep* and *wish list*.

The affinity diagram resulted in the three tables 4.3, 4.4 and 4.5, which contain conclusions and interpreted requirements. The requirements were later used during the sketching in iteration zero, see section 5.1.1.

**Table 4.3:** Conclusions regarding the persons from the affinity diagram.

<i>ID</i>	<i>Description</i>
A1.1	Tom: Have been employed five years and is an advisor and a team leader.
A1.2	Harald: Have been employed six months and is head of advisor.
A1.3	Simon: Have been employed one year and is an advisor.
A2.0	Team leaders are also advisors, which make them request the same functionality as the advisors, but need another layer of information.
A3.0	Advisors are using Partner for almost all tasks in their workday.

**Table 4.4:** Conclusions about Partner elicited from the affinity diagram.

<i>ID</i>	<i>Description</i>
P1.0	Partner has all the information that is needed, but unfortunately it is not feasible to find it since Partner is too comprehensive. <i>Partner is mainly used for:</i>
P2.1	Customer troubleshooting.
P2.2	Looking up customers.
P2.3	Structuring the workday with emails and tickets.
P3.0	It is slow.
P4.0	In some cases, the design is interpreted as unpleasant.

**Table 4.5:** Conclusions and requirements regarding the dashboard elicited from the affinity diagram. The requirements are not ranked and sorted in any specific order.

<i>ID</i>	<i>Description</i>
<i>Goals</i>	
D1.1	Create an overview for today's agenda and the advisor's customers.
D1.2	Be able to use it proactive.
D1.3	Relevant information.
D1.4	Appealing user interface.
D1.5	Short loading times.
D1.6	Must be an overview, not be used as the tool accomplishing the tasks.
<i>Features to keep</i>	
D2.1	The tickets functionality, but sometimes the color scheme is not evident.
D2.2	Listing of ongoing campaigns.
<i>Wish list</i>	
<b>Advisors</b>	
D3.1	Keep track on your customers, to be able to work proactive.
D3.2	Different layers where content for head of advisors, team leaders and advisors differs.
D3.3	Better overview of the mailtickets and messages.
D3.4	Shortcuts to tools which are often used, but are located far away from the dashboard.
D3.5	Show soon ending contracts.
D3.6	Add a calendar.
<b>Extra add-ons for team leader</b>	
D3.7	Information about the team.
<b>Extra add-ons for head of advisors</b>	
D3.8	More statistics, especially sale statistics.

### 4.5.3 User persona

From the interviews and the affinity diagram two user personas were developed. Due to the discovery of different roles the need for two user personas, one primary

and one secondary, was necessary.

The two personas, found in appendix B, were developed using the guidelines described in section 2.1.4 under *User persona*.

The first persona, called Anna Almér, is the primary persona. She is an advisor, and about one year experience working at Telavox. She has worked in previous customer support environments before and is used to complex systems. This user persona is directly anchored to the primary user where our project is focused.

The secondary persona, Simon Hammarström, also has the role of advisor, but simultaneously is a team leader. This affects some details in what is wanted based on observation and interviews. It can also be of help when prioritizing functions and design that can be aimed specifically towards the team leader role, and not be of particular use for an advisor. The user persona are to distinguish the extension the role of team leader has.

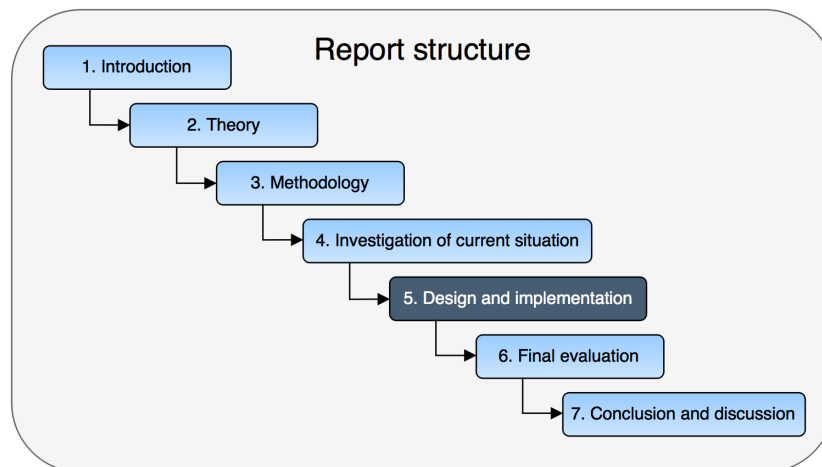
---

## Design and implementation

---

In this chapter the implementation phase is described and results from the various iterations are presented. It consists of the first iteration called "Iteration 0", where sketches and low-fidelity prototypes were developed and resulted in a high-fidelity prototype. The other iterations was built on the previous iteration to implement functionality and new design decisions. In this chapter, RQ3 (see section 1.3) will be answered.

In figure 5.1, one can see the part where this chapter belongs in the report structure.



**Figure 5.1:** The report structure with current chapter highlighted.

## 5.1 Iteration 0: Prototype phase

### 5.1.1 Low-fidelity prototype

The sketches were done during a period of three days. This was to develop new ideas and concepts that was not thought of, and to avoid being creatively exhausted. Carl came up with ten different sketches done during these three days. Johan spent only one day to focus on sketching and produced two sketches. All the hand drawn sketches can be viewed in appendix C.1.

Every sketch had one starting-point, the header of Partner. This thesis focus on the dashboard, and not to implement new functions or change the header. Therefore the header stayed the same in every sketch.

All sketches had their base in the requirements in table 4.5, and was created with the theory in section 2.2.3 in mind. An example of how visual noise was prevented is the tab functionality which was present in several sketches. Moreover, in the sketching stage, some of the seven mentioned concepts in section 2.2.1 were considered, but in the low stage some things such as detailed affordances, signifiers and constraints were hard to get across. Feedback was things that was noted but not sketched, and the main focus was on the mapping and conceptual model.

The sketches where then analyzed in a meeting, where positive and negative features were discussed and evaluated. When the features had been discussed a new sketch was drawn together on a white board. Things like placement, mapping, size, affordances and signifiers were discussed, and three new sketches emerged, that can be viewed in appendix C.2.

The sketches were then drawn in the program draw.io to create mock ups to show stakeholders. The three mock ups, seen in figure 5.2, 5.3 and 5.4 are the result from the sketch meeting. Central from our investigation of Dashboard 1.0 was that the ticket function used on the dashboard was used frequently as seen by both interviews and quantitative data. This is reflected in all three mock ups, where the ticket tool is placed in the same spot for every mock up with slight changes in functionality described in the sections below.

The first mock up has, as stated, the ticket tool placed in the far left corner. The ticket system is color coded to represent open tickets (red) and awaiting tickets (blue). The ticket bars are clickable, and if clicked a text of the latest information from that ticket is presented. Under the ticket tool a status bar is shown to indicate how many of the advisor's tickets that are open, awaiting and hidden. To the right, a list of all the advisor's customers is presented, and below is a statistical overview of the teams that work in the advisor organization. The last tools are a search function with various unspecified settings to list tickets and a calendar in the bottom. The placement is based of what could be grouped as the *advisor's own tools* and *general tools*. The tickets and customers that are



the advisor's own are therefore placed on the top, and the other general tools are placed below.

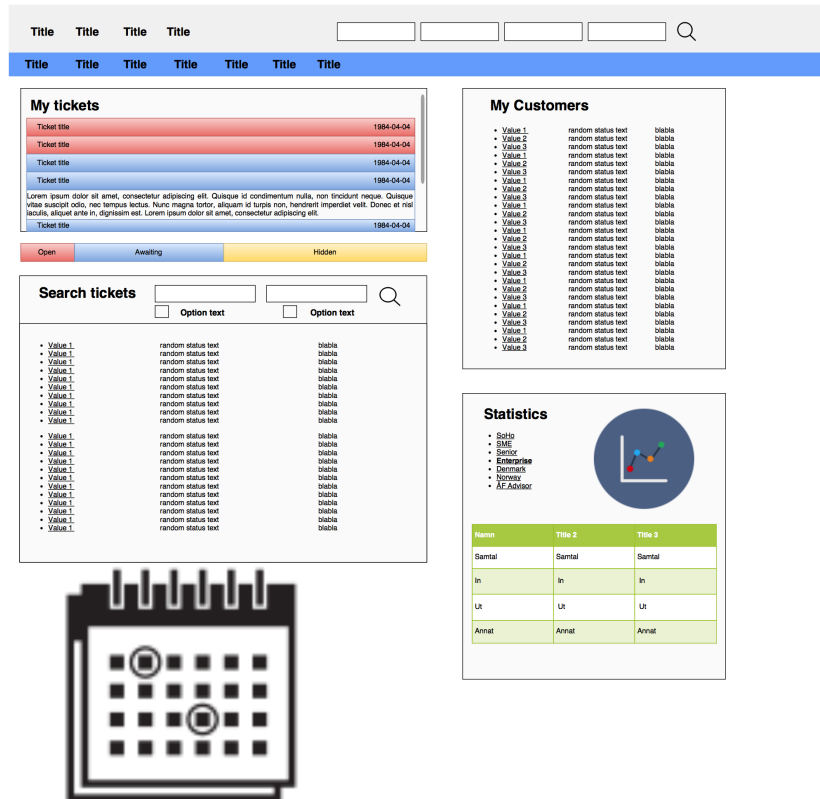


Figure 5.2: The first mock up.

The second mock up uses a slightly modified ticket system, with a tab system that can show ones own, team members and other tickets. The same clickable feature is found, but the color coding is gone. To the right a statistics field, unspecified, can be found to display important analytical data like in and outgoing calls. Down at the bottom a *customer proactive* tool can be found, displaying customers that has filled some criteria that indicate that something might be wrong in a red box. For example their contract is coming to an end, or other indications that they might be having a bad experience with the service. This is to accommodate the need to work in a proactive way. Below is a list of information for all of the advisors customers.

The third mock up is similar to the first one in the ticket and customer list, but has has two main differences. The calendar gets a more central role and is placed more in the center, and besides it there is a tab menu that contains various

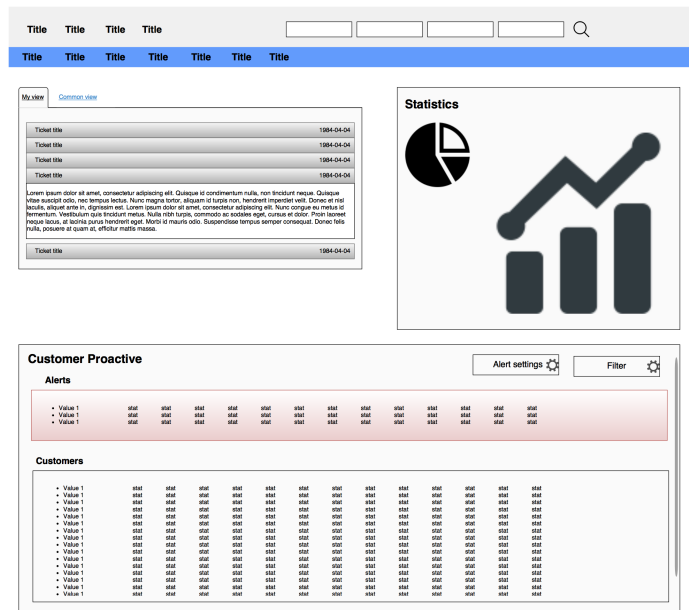


Figure 5.3: The second mock up.

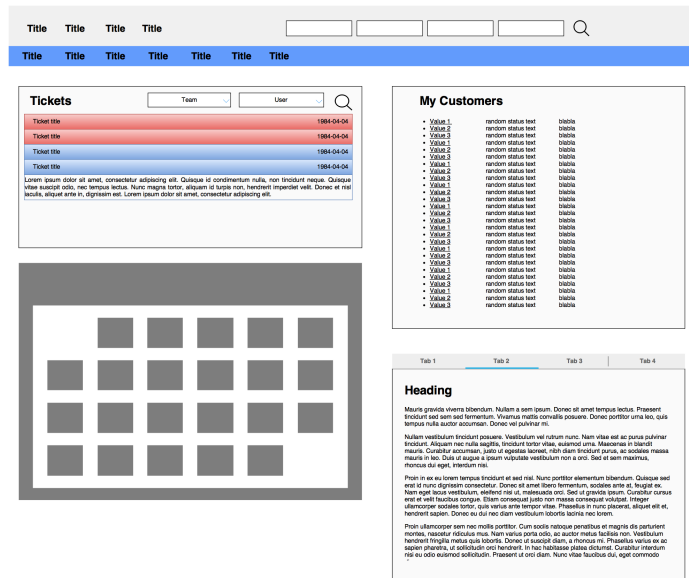


Figure 5.4: The third mock up.

information such as document links, statistics, latest customers and various information that can vary in relevance to an advisor. The motivation is to save space but also keep the noise to a minimum but still having the information there if it is needed.

The stakeholder meeting consisted of two of the previous three interview subjects. It was a more informal meeting to present and talk about the sketches and to validate the interview results. The interview lasted 45 minutes and was done with a computer displaying the three mock ups while they were presented. After the presentation the two subjects got to voice their thoughts regarding the mock ups.

There was a very positive response to the second mock up. The tool to see and react to the proactive alerts were a "must-have". They agreed on the importance of the ticket bar, and felt that the functionality to click and view the most recent message was a good feature. To view different teams in other tabs was also good.

Regarding things in the mock up that had a lot of information, as the customer list in the first and third mock up or the general status of customers for the proactive tool in the second mock up, the response was that there is too much information and not any way to process it all.

They also pointed out existing things that were missing in the mock ups that was of use in the current dashboard. To be able to view the current news, campaigns, latest created customers and quotations is of great importance to have an overview of which was missing.

With the feedback from the meeting, a new mock up was created, which was the one that was used for developing the high-fidelity prototype. This mock up can be seen in figure 5.5

### 5.1.2 High-fidelity prototype

The last low-fidelity prototype made it possible to start developing the high-fidelity prototype, which later became the base of the product. The Partner repository was cloned and a new branch in the repository was created. Further, a blank dashboard2.jsp was generated and from there the new prototype was developed.

The design decisions were based on the requirements found in the table 4.5. Most of these goals were central for the advisor work flow and the scope of the thesis, and therefore included. Below are the justifications of how the requirements were covered.

#### Regarding D1.4

To satisfy the request for a more appealing user interface, Bootstrap was used. To begin with, the front end was developed based on the latest

low-fidelity prototype. Instantly, a problem with the implementation of Bootstrap was discovered. Unfortunately, the CSS stylesheet for Bootstrap changed the interface of the original Partner page, including the header. This led to a discussion whether it was possible to do a work around, create a completely new CSS stylesheet or if it was best to edit Bootstrap's stylesheet. The solution ended up in an easy work around, where the stylesheet for the Partner template was the last thing included in the DOM. This made it override Bootstrap's stylesheet.

### Regarding D2.1, D3.3

As mentioned earlier, the ticket system was often used by the advisors, and therefore a vital functionality to implement. To get the overview of the tickets and messages a preview of the latest note of each ticket was added.

### Regarding D2.2, D3.4

These requirements was handled by adding the general info section.

### Regarding D1.2, D3.1, D3.5

The three requirements were all covered by the customer proactive view.

### Regarding D3.6

This requirement was covered by adding the Google calendar section.

### Regarding D1.1, D1.3 and D1.6

All these requirements and goals are regarding getting an overview. It is hard to concretize exactly what has been done to satisfy these requirements, but the project overall is about creating a dashboard for the advisors, thus

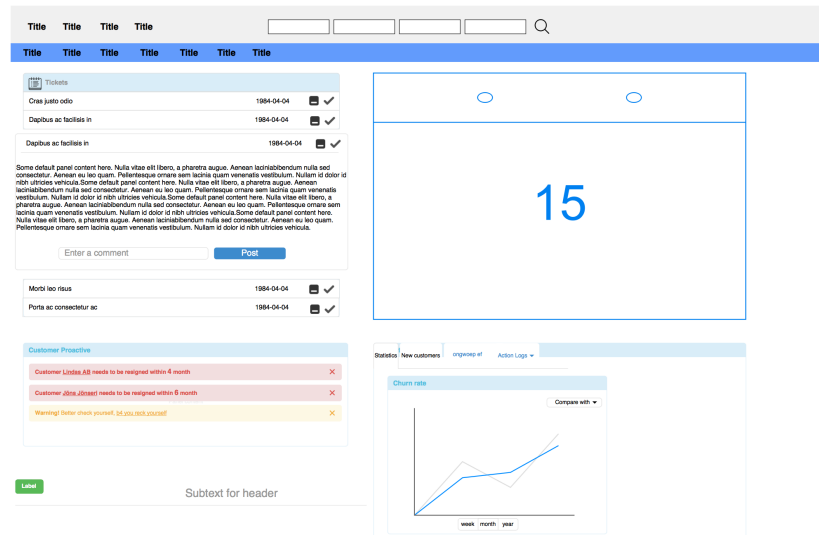


Figure 5.5: The final mock up.

these are all covered by the four sections together.

The requirements that were not covered were: **D1.5**, **D3.2**, **D3.7** and **D3.8**. The requirement D1.5 was not covered as it was only a prototype. The other requirements were out of the scope of this thesis, and therefore not included.

After three days, a high-fidelity prototype was finished, which made it possible to explore the dashboard, including some hard coded click-functionality. The result of the prototype was saved in a branch for later studies. The prototype can be found in figure 5.6.

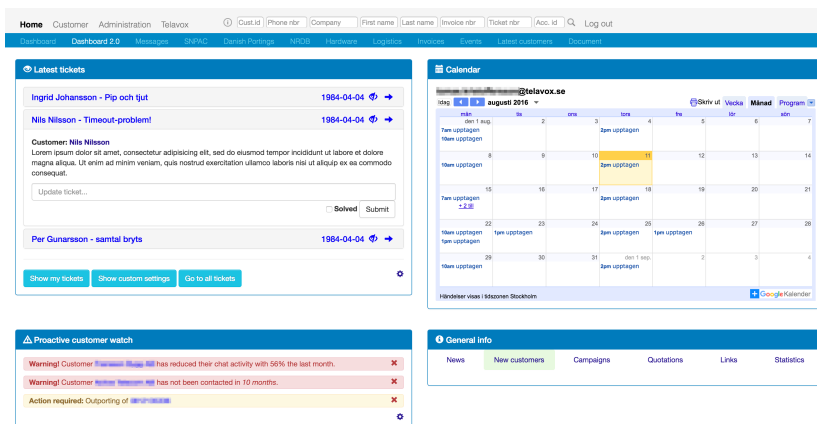


Figure 5.6: The high-fidelity prototype.

## 5.2 Iteration 1: First product release

Iteration 1 continued the product coding development. It was natural to continue developing the high-fidelity prototype, since it was a good base for the product. The iteration was mostly about creating and developing the improved ticket presentation, which was the functionality that was used most according to the previous investigation phase. Therefore, the decision to begin with the tickets was evident. However, the calendar integration was quite easy to implement, using an iframe, and was therefore also implemented in this phase.

In the iteration some particular interesting design decisions were made. In the following sections they are mentioned. The result of the first iteration can be seen in figure 5.7.

When the users clicked a ticket in the list, an accordion effect opened the latest ticket information. Unfortunately, the latest ticket information was often just a status message, lacking information about the actual ticket. To solve this, all the latest messages, with the same time stamp, was merged into the same message.

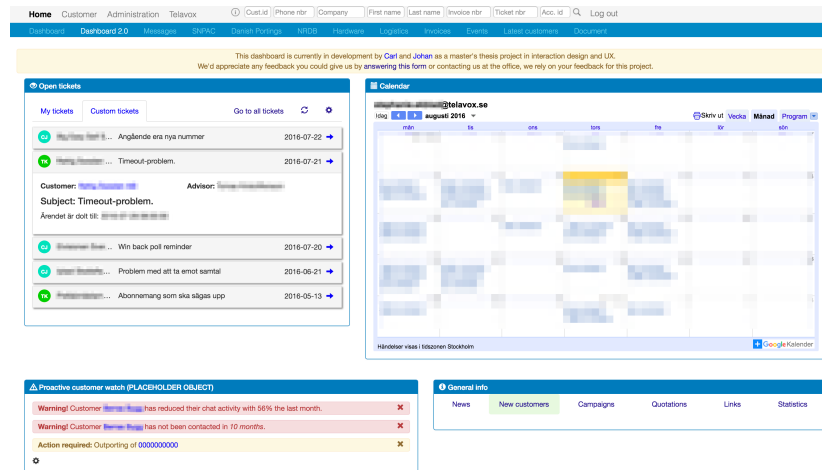


Figure 5.7: The result of the first iteration.

Moreover, the accordion effect was provided by the framework Bootstrap, which had some compatibility problems with a legacy framework, so to get it function properly the legacy framework had to be excluded.

At first, all tickets for the signed in user were showed in the list of tickets. But as mentioned in the theory (section 2.2.3), do not include visual noise. Therefore, it was decided to only view the open tickets for the user. Later, during discussions with advisors, it was clear that advisors needed to see their colleagues' tickets as well, due to need of getting an picture of other advisors' work load. To solve this, a second tab view was developed, where the advisor could choose which advisors' tickets to view. This view was called "Custom tickets view".

These two views had to be easy toggling between, thus two buttons were created, one for the signed in user and one for the custom tickets view. However, this method was functional, but not good enough, since the tickets had to be loaded from the database and into the DOM every time the user changed view. This is both performance-intensive and does not support the functionality to toggle fast between them, which would decrease the user flow and experience. Instead the buttons were changed to toggleable tabs, where the content of each view were loaded into the DOM asynchronous, and also left there until it was manually updated. This enabled the user to switch between the views quick and keep the settings for each view.

Consequently, cookies for the dashboard were introduced. Those were used for keeping the information about which view that was latest active and also which advisors that should show up in the custom tickets view.

When switching between the views, there were some waiting time to get the response from the server. To create a better user experience, a spinner loader

was added to the updating view to add feedback to the user. In this way, the user knew that the view was fetching data. The spinner was similarly added when a user clicked any link that forwarded the user to another page in Partner.

In the custom ticket view, the need to distinguish the various advisors' tickets arose. This was solved by implementing a colored circle with the assigned advisor's initials. Every advisor got a unique color to help identify different advisors' tickets, and the circle was placed on the far left of every ticket header. The color was chosen to be within a range that was readable together with a white text, and was generated based on the advisor name. This feature was inspired from various established services such as Dropbox and Google.

To let the user choose of which advisor to show in the customer tickets view, a tree view of buttons representing all the advisors was created. This feature was classified as a custom setting, and therefore not necessary to keep constantly visible. For that reason, it was placed in a modal, that could be opened when needed. The button for this modal was represented by a cog, which is well known as a settings icon. This icon, together with an update button and "Go to all tickets" was placed in the upper right corner of the ticket section to separate them from the tabs functionality.

In the mentioned modal the tree view of the advisors was created and to indicate that the groups could be expanded, a plus icon was placed on the right of the group. When the group was expanded the plus icon changed to a minus icon to show that the group was expanded.

The button, representing each advisor, had two modes, activated and deactivated. The two modes were indicated by two different colors and could be switched to the other by click the button. It was also possible to activate a whole group by clicking the group button. Moreover, to make the tree view consistent it was important to make sure the all the advisor buttons were synced to the group button. For example, when all of the advisor buttons in a group were activated, the group button should be activated automatically. Similarly, if a user activated a whole group by the group button, and later decided to deactivate one of the advisors in the group, the group button should also be deactivated automatically.

In this iteration the requirement **D1.5** (see table 4.5) was implemented. This requirement was put in focus, and thought of in every stage, both regarding developing front- and back end functionality. Requirement **D3.4** was less prioritized to implement, and was therefore removed from the dashboard.

### 5.2.1 Evaluation

The first and functional draft of the Dashboard 2.0 was put besides the Dashboard 1.0 in the menu in Partner. This made it easy for the advisors to toggle between the dashboards and also compare them. On the top of Dashboard 2.0, an

information text was placed to inform the advisors that this was the dashboard in development by Carl and Johan as a master's thesis project in interaction design and UX. Moreover, there was a link to a Google Form, asking the advisors to submit any possible feedback they could come up with. To clearly announce that Dashboard 2.0 was available, an email was sent out to all the advisors stating the same information.

The form contained two free text fields questions asking about good things about Dashboard 2.0 and elements that needed improvement. This was to enable the advisor to write as much feedback as wanted. Unfortunately, the form only got two answers. Instead, more specific feedback was received by asking the advisors in first person. The feedback collected, both with the form and by first person, were compiled to:

### **Good**

- G1** Quick and much better ticket system. Feels up-to-date.
- G2** Clear overview.
- G3** The calendar integration is great.

### **Need improvement**

- NI1** *General info* section would fit better over the calendar.
- NI2** *Proactive watch* section is better placed on the top, besides the ticket section.
- NI3** Would be good with an answer-functionality in the ticket system.
- NI4** The four sections are better when they have symmetric size.
- NI5** The *ticket section* feels overwhelming.
- NI6** It is not clear what the plus and minus signs in the modal tree view means.
- NI7** It is not possible to see tickets that is unassigned.

## **5.3 Iteration 2: Adding functionality and new features**

With the feedback from iteration one, it was natural to initiate iteration two. The feedback became a good reference for a To-Do-list, where each item was a motive for what and how to change the Dashboard 2.0. The development of iteration two began with changing the things that had been requested in the feedback. The result of the second iteration can be seen in figure 5.8.



The screenshot displays the Dashboard 2.0 interface. At the top, there is a navigation bar with 'Home' and 'Customer Administration Telavox'. Below this is a search bar with fields for 'Cust.id', 'Phone nbr', 'Company', 'First name', 'Last name', 'Invoice nbr', 'Ticket nbr', and 'Acc. id'. The main content area is divided into four sections:

- Open tickets:** Shows a list of tickets with columns for 'My tickets', 'Custom tickets', and 'Go to all tickets'. A detailed view of a ticket is shown below, including fields for 'Customer', 'Advisor', 'Subject', and a date 'Ärendet är dolt till: 2016-09-11 07:00:00'. A 'Submit' button is present.
- Proactive customer watch (PLACEHOLDER OBJECT):** Displays a list of proactive watch items with warning and feedback messages. Examples include: 'Warning! Customer Berras Bygg has reduced their chat activity with 56% the last month.', 'Warning! Customer Berras Bygg has not been contacted in 10 months.', 'Warning! Customer Berras Bygg has created 25 new tickets regarding technical problem in the last 4 weeks.', 'Action required: Outporting of [blurred] for customer Berras Bygg.', 'Action required: Renew contract with [blurred] Expires 2016-10-31.', 'Action required: Outporting of [blurred] for customer Berras Bygg.', 'Action required: Outporting of [blurred] for customer Berras Bygg.', 'Action required: Incoming porting of [blurred] for customer Telavox Tomas Tak.', 'Feedback: Customer [blurred] has increased the chat activity by 20% the last month.'
- Calendar:** Shows a calendar for September 2016 with a grid of dates and event counts. For example, Monday 29th has '+7 more', Tuesday 30th has '+4 more', Wednesday 31st has '+4 more', Thursday 1st has '+3 more', and Friday 2nd has '+5 more'.
- General info:** Displays a table with columns for 'Date', 'Quantity', 'Company', and 'Type'. The data rows show dates like '2016-09-30 08:00', '2016-10-05 08:00', '2016-10-28 08:00', and '2016-11-24 08:00' with corresponding quantities and types.

Figure 5.8: Dashboard 2.0. Blurred parts where sensitive information could be presented

### Decisions regarding NI1 and NI2

Both the requests, NI1 and NI2, were regarding the placement of the two elements, where both elements were requested to be on the top. Since the ticket section was not questionable to move, the conflict led to a discussion about which section that was more important. However, in the end the proactive watch was placed on the top, because of its importance of first hand attention. Consequently, the calendar was placed in the bottom left corner. This created a mapping where the advisor's user information was placed at the top, and more general information was placed on the lower sections, making them mapped to each top and bottom section.

### Decisions regarding NI3

This was discussed during the first iteration, but never implemented due to lack of time. However, in iteration two there was enough time and was implemented with a small text input and a submit button. In the main ticket system provided by Partner there were several possibilities to submit different kinds of notes, i.e. internal note and answer the customer. Since the requirement D1.6 (table 4.5, section 4.5.2) said that Dashboard 2.0 should not be used to accomplish tasks, the best solution was to

only add the internal note functionality. However, if the full ticket note functionality was requested later, it was possible to add it on demand. The input text field was placed under the latest notes of the ticket and the possibility to submit the ticket by pressing the key return was added. When the note was submitted with success, the note was also added dynamically under the latest note in the ticket. To make it clear it was newly created a green fading background effect was added. Since the ticket also was the most recently changed the ticket was moved to the top of the list of tickets.

#### **Decisions regarding NI4**

This request were not discussed much, it seemed more logic to have four symmetric sections in the dashboard. It was changed as requested.

#### **Decisions regarding NI5**

At first this was hard to understand, but after a closer look at the requesting advisor's screen, it was clear that for the advisor, Dashboard 2.0 was not viewed in the same way as in our test environment. The margin between the tickets in the ticket section was too tight in the advisor's browser. This was probably because of the differences in running Dashboard 2.0 on Microsoft Windows compared to Apple OS X, since both used Google Chrome as browser. However, to solve this some extra margin was added between the tickets in the list.

#### **Decisions regarding NI6 and NI7**

These two requests were not implemented. The main reason **NI6** was not implemented was that it was not regarded as serious as it actually was. The request **NI7** was not chosen to be implemented due to lack of time and to not make the tool a replacement of the current ticket system that is located deeper in to Partner.

Apart from the changes regarding the feedback there were several improvements done in the second iteration. The ticket system was equipped with a functionality to alert the advisor about tickets that had not been updated in the last 24 hours. This was mainly advertised with a pink background color of the specific ticket, but also with an information text in the bottom of the ticket view. As soon the ticket was updated, either by submitting an internal note or solving the ticket, the background went back to its default color.

Moreover, the *General info* section was improved. In the Home tab, latest news and campaigns were included. This was to satisfy requirement D1.3 (table 4.5). Further, a table of upcoming porting's for the signed in advisor was showed in the second tab to let the advisor work proactive (D1.2, table 4.5). In the last tab (My deliveries), a tool previous built for Partner was included to create the possibility for the advisor to work more proactive. The tool showed the upcoming deliveries for the advisor.

In general, all the new information was loaded with AJAX to lower the loading times, according to requirement D1.5 (table 4.5).

The evaluation of the result of the second iteration was made more in detail and is presented in chapter 6.

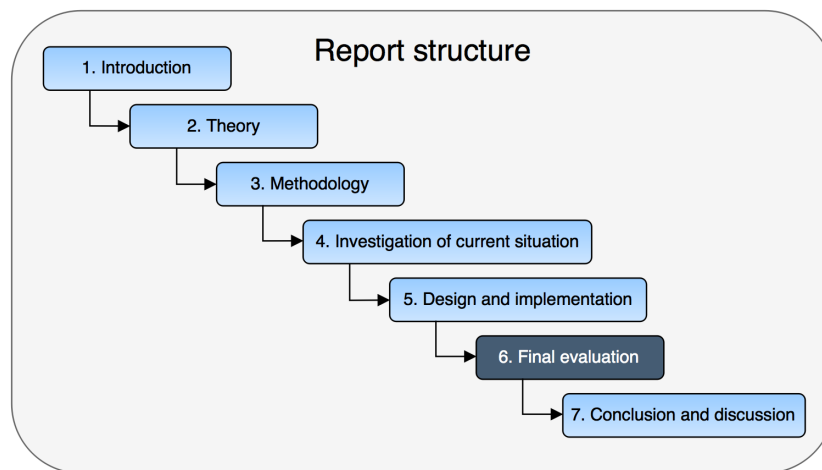
---

## Final evaluation

---

This chapter presents evaluation of the results obtained in the previous parts of the report. It covers three analysis methods of the dashboard: a design analysis and comparison of Dashboard 1.0 and Dashboard 2.0; an expert evaluation of Dashboard 2.0; and finally user testing. In this chapter, RQ4 (see section 1.3) will be answered.

In figure 6.1 one can see the part where this chapter belongs in the report structure.



**Figure 6.1:** The report structure with current chapter highlighted.

### 6.1 Design analysis introduction

During the interviews certain characteristics to look for in a site emerged. The characteristics that were regarded during this analysis was based on the

requirements in table 4.5.

Some of these characteristics were measurable, such as speed, but most were subjective. In the analysis the main focus was on subjective thoughts about the design.

### 6.1.1 Analysis of Dashboard 1.0

The old dashboard, as can be seen in figure 1.1, provides two columns with various rows of information. The general look is based on white and blue with shades of gray and sometimes red to emphasize urgency. All interactions with links redirects one to another page, and the tool is clearly to display general and as much information as possible. Mostly everything is built using HTML table elements, and some icons are presented, as smileys to indicate good and bad conditions. It is not presented in which states and when they change.

The first column provides a retailer login function, latest news, ticket information (a maximum of seven), porting information for all of Telavox and a document list with the latest added documents. The second column provides information to the advisor about own quotations, campaign information, latest created customer, order value for the month and graphs. All information in the columns are placed on top of each other and in no regard of screen size from what can be found on tests. The two columns have a fixed size.

The relevance of the information presented varies. In the quantitative analysis in section 4.1.2 the ticket tool is used to great extent. It only displays seven tickets at maximum, and can only display the logged in advisor's tickets. Else one has to move to the ticket site with the "Show all" link.

The other content can be of interest, but is presented together with other general information. Outgoing and incoming portings all display the state for all of Telavox, not the advisor's specific customers. Both the order value and graphs are of little relevance to the advisors, and are unnecessary.

To sum up, the page is a place where a lot of information is presented, some relevant and some not. Sometimes important information is cut (as with the tickets to only display seven) or too much information is presented as in the case of portings. The features seem to be placed from top to bottom of the site and added in an ad hoc way when thought of. The interaction with elements on the site is lackluster, and the only direct interaction as a user is the basic underline of text when hovering above a link.

### 6.1.2 Analysis of Dashboard 2.0

The new dashboard, as can be seen in figure 5.8, is clearly divided into four sections equally distributed over the screen. The sections are grouped by panels

with a header describing the panel and a section with the content. The sections are responsive to the size of the screen and at a smaller screen size the sections are placed on top of each other, instead of in two columns.

The links have no hover underline, and only a small color change occurs when the link is visited. Outgoing links to pages with a large load time triggers a spinner to indicate that a loading session is in progress. This is also implemented in the internal loading sessions in the site.

The first part of the site, the “My ticket” view displays the open tickets for the logged in advisor in one tab, and one tab that displays manually entered advisors’ open tickets. The tickets are dynamically loaded into the site, and can be loaded without reloading the page. This functionality is present in all of Dashboard 2.0 when it is not necessary to move outside the dashboard.

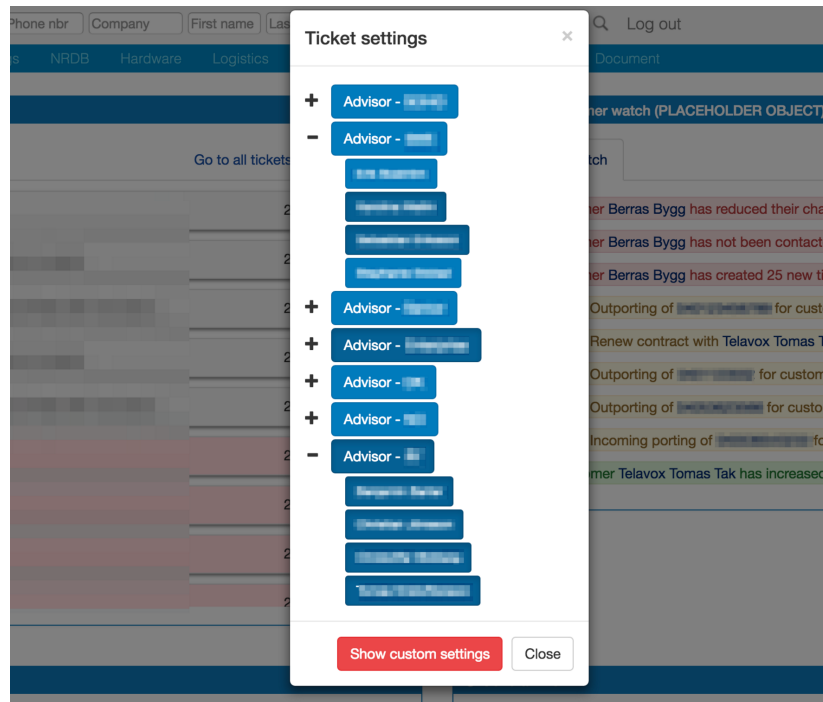
The tickets have an indicator to display which advisor it has been assigned to with a fixed generated color and initials of the advisor. Moreover, they have a text with the customer that the ticket is concerning, ticket subject and the time it was last updated. A ticket that has been updated in the last 24 hours has a grey background, and a ticket that has not been attended for over 24 hours has a red background to signify that there is a need to handle this ticket. A shadow on the ticket status is present to indicate that it is clickable.

When clicked the ticket drops down with an accordion effect, presenting information about who the customer is (with a direct link to them), the advisors full name, the subject and the last updated note on the ticket. A text box for submitting quick internal messages and update the ticket is present at the bottom. When an internal comment is sent the ticket is updated, placed at the top of the ticket list to mark the ticket updated, and the note is placed inside the ticket view.

The tabs are made to be updatable without refreshing the page. To do this a refresh symbol is placed in the tab menu to the right together with a link to the full ticket system. To the far right there is a settings wheel to enter the advisors searched on in the “Custom tickets”-tab.

When entering the settings, a modal pops on the screen, with the focus on what team or advisor the user wants to view, as seen in figure 6.2. The advisors are grouped up in teams, and it is possible to select teams and sole advisors with a tree structure. The selected and unselected advisors or teams are both marked with a blue color, and the color distinction is not very clear. When selecting “Show custom settings” the view is updated with the selected advisors.

In the top right corner a “Proactive customer watch” functionality is present, that warns an advisor about several crucial events that has happened. For example it warns the advisor with one note that a customer needs to have its contract renewed or that their chat activity has decreased. The colors green, yellow and red were chosen to represent the severity scale, green being the lowest and red being the highest. A link to the concerned customer is provided in the



**Figure 6.2:** View of the advisor menu in Dashboard 2.0. Blurred parts where sensitive information could be presented

warning, and when the advisor finds it handled, it can simply discard the note by clicking the x on the right side.

The bottom left contains a Google calendar section that uses the advisors work email to display their schedule. It is a Google calendar iframe, and works good with the color scheme of the page in general.

The bottom right of the page contains a tab view that contains various kinds of information. This is used to put all general information in one place, easy to navigate when needed. The first tab contains the latest news and campaign information, the second is a table display of the advisor's portings and the last tab contains the functionality "My deliveries", that is loaded from another page in Partner and not designed during this project.

### 6.1.3 Comparison of the two dashboards

Comparing the design of the two dashboards, the information is gathered in a more structured way in 2.0. Instead of adding things ad hoc, the information on the site is gathered in four specific sections. The information is customized for

the advisor using it, which is to benefit the work flow and focus, and is lacking in Dashboard 1.0. Relevance and usage is also prioritized, and removes redundant information in the new dashboard.

Dashboard 2.0 is more complex with settings, tab views and calendar integration which can be a lot to navigate around compared to 1.0, where the information is presented from top to bottom. This can be of negative value in the beginning due to hesitation to change, test and use every function on the live database (to not break anything in the Partner system).

While comparing the two designs the only design element that stands out in Dashboard 1.0 is the use of emojis and table row highlight. Dashboard 2.0 uses familiar design elements from Bootstrap, gives user feedback, uses asynchronous loading, and color coding that is based on existing conventions from various sites.

Dashboard 2.0 differs a lot from Dashboard 1.0 and is a more specific tool designed to help the individual advisor get an overview. Dashboard 1.0 was developed when different functionality was needed, while the Dashboard 2.0 tries to satisfy the advisor needs.

## 6.2 Expert evaluation: UX-team at Telavox

To get a professional point of way, a meeting with the UX-team at Telavox was set up. All three people from the team were present at the meeting. The meeting was informal, where the Dashboard 2.0 and its elements first were presented, and continued with a discussion of their thoughts. Basically one question was asked: “What do you think about Dashboard 2.0?”

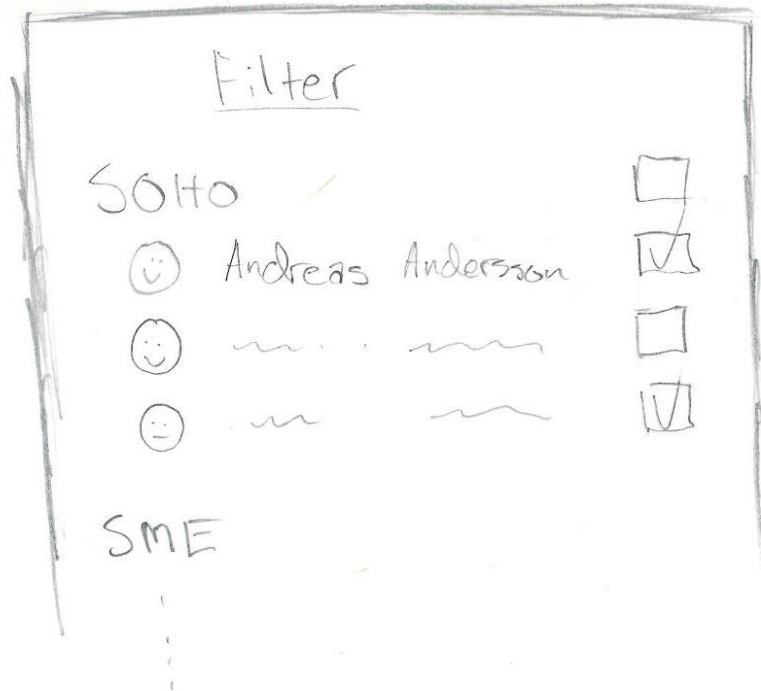
The general thought about Dashboard 2.0 was that it has been improved a lot. The purpose of this tool is help the advisors, therefore usability and readability are the most important things to consider. Their thoughts and tips was summarized as follows:

- In the ticket system, instead of the color circles with the assigned advisor’s initials, the avatar in the Telavox database of respectively advisor should be implemented.
- The settings cog in the ticket system, should be changed to a filter icon instead, because it is a filter-functionality, not settings.
- Isolated icons usually means that it is clickable.
- There should be a divider between a ticket’s date and go-to-ticket-arrow, otherwise they might be associated with each other.

There were several thoughts about the tree view for the custom ticket settings view, here seen in figure 6.2. First of all, it was not clear that a user could



expand the tree. It was hard to see the difference between a group and an advisor. Together with UX-team a new sketch of the filter was developed. The sketch can be seen in figure 6.3. Main focus was to separate the group categories from the members of a team and change the buttons to checkboxes.



**Figure 6.3:** Sketch of the filter settings developed during the expert evaluation with the UX-team at Telavox.

Moreover, general advice, i.e. sketching with every element, even in the implemented design, was given. This was to quickly see if something works, and to reduce the cost of seeing if something works. Finally, they all came up with one advise for future work, “Remove as much stuff as possible”. This was to increase function value and to focus on what was important.

### 6.3 User testing

The last part of the evaluation phase was user testing. Three advisors, from three different segments, were randomly asked to participate. This was to make the tests independent and to ensure a broad perspective. However, it was important the test participants were all advisors, and not a team leader or another higher

manager, because of the delimitations the project had.

The test was set up in a test environment, with a test database, so the advisor felt secure not breaking anything while trying out Dashboard 2.0. However, it was of importance that the advisor could use its own computer and workstation, with its own username and password to make the test as genuine, for the advisor, as possible. Moreover, the advisor were told to think out loud during the test, to ensure nothing was left out (see 2.4.2).

Each test began with an introduction where the purpose and agenda of the test were explained. The test itself, consisted of two parts. The first was to get the advisor familiar with Dashboard 2.0, where the advisor got to explore the tool by itself and ask questions if there were any uncertainties. The second part contained three scenarios or tasks (see 2.4.2), which the advisor got to solve by best of their ability. In this way it was possible to see if the advisor had understood Dashboard 2.0 and if the dashboard had been designed in an intuitive way.

The three scenarios were chosen to cover most of the functionality on Dashboard 2.0. Therefore, it was natural to have one question for each of the four sections: ticket system, proactive watch, calendar and general info. Since the calendar, was not designed in this project, and it is self-evident, it was left out. The other scenarios were:

**Scenario A** Your colleague, “One team member”, is ill and need some backup with two of his tickets. Choose the oldest of his tickets and place an internal note on it.

**Scenario B** Are you able to see if you have any upcoming portings or deliveries?

**Scenario C** Are you able to make any conclusions of the information in the proactive watch section?

Scenario A was chosen to see if the advisor had understood the ticket system. In this scenario the advisor had to go to the custom tickets view, open the modal with the settings, expand the tree, choose the mentioned colleague, close the modal, find the oldest ticket and place an internal note on that one. The deem this task successful, it should be successfully completed within one minute from the task was given.

Scenario B was chosen to see if the advisor could find the views of the requested information. The advisor had to click on the respectively view in the general info section to succeed with the task.

Scenario C was chosen to see if the information in the proactive watch section was clear enough to draw conclusions about two customers' satisfaction. One of the customers had not been contacted in ten months, had reduced their chat activity with 56 percent the last month and had created 25 new tickets regarding technical problem in the last four weeks. The intention was for the advisor to see that Telavox was about to lose the customer. The other customer had positive

feedback about its chat activity, and the intention was for the advisor to see that the customer was satisfied at the moment.

The test ended with the question: “Does this tool make you get a better overview of your day?”. All the results from the tests were of a qualitative nature based on the testers’ experiences. Where quantitative data, such as elapsed time, could be applied, the thesis workers instead chose to explain the functionality when the tester got stuck on a scenario. Time was noted during the tests, but was never part of the result, due to its irrelevance, as it never exceeded a reasonable time limit, which was considered to be around one minute.

## Result

The first advisor had no problem understanding Dashboard 2.0 in general, so no explaining of the functionality was needed. Other observations regarding the scenarios were:

**Scenario A** was no problem understanding and complete within a reasonable time. However, the advisor chose to select the whole team’s tickets and find the oldest manually instead. This was because the advisor did not understand that the teams in the settings were expandable. As mentioned earlier, our intention was for the advisor to expand the tree.

**Scenario B** was completed without hesitation. However, the advisor did comment that since his team only managed the retailers and not end customers, the porting and deliveries tabs were unnecessary, since the retailers were managing that themselves.

**Scenario C** was completed without any problems. The advisor understood that one customer was not satisfied and the other was satisfied.

In general, the advisor was pleased and thought the ticket system was the greatest improvement of Dashboard 2.0 compared to Dashboard 1.0.

The second advisor was not familiar with Dashboard 2.0, and the walkthrough of the different elements was needed. The first spontaneous thought of the advisor was that the calendar was great to have on the dashboard.

**Scenario A** was completed within a reasonable time. However, this was after the walkthrough of the functionality of Dashboard 2.0. The advisor did not understand that the tickets were expandable, but had no problem understanding the arrow “Go to ticket”.

**Scenario B** was completed without hesitation. The content of the General info section was clear and easy to access. The My portings tab was especially spoken for.

**Scenario C** was not completed. The color coding of the importance was clear, but no general conclusion about the two customers was made.

In general, the advisor was satisfied and had only positive feedback. Dashboard 2.0 would give a better overview of today's tasks.

The last advisor was not familiar with Dashboard 2.0, thus the walkthrough was necessary again. At first sight of Dashboard 2.0, the calendar was highly appreciated.

**Scenario A** was completed in a reasonable timely manner and the advisor thought the new ticket system was quick and responsive. But unfortunately the color identifier with initials for which advisor the ticket was assigned to was not clear enough. Therefore, a hint was needed for this scenario to be completed.

**Scenario B** was completed without hesitation. My portings was especially usable, and created a good overview. The deliveries tab was condemned, due to the need of scrolling in two dimensions.

**Scenario C** was completed with the correct conclusion. The color coding of the messages was clear, and the red ones was prioritized.

In general, the advisor was satisfied with Dashboard 2.0 and thought it would create a better overview of the daily tasks. The advisor also felt it was good that these features were gathered in one place.

## User testing conclusion

From these three tests following conclusions were made regarding the four different areas:

### Ticket system

- The ticket system section was highly appreciated of all participators.
- The tree view of the custom ticket settings were beneath contempt, no one of the three participators understood it completely, instead it created confusion.
- To make this ticket system usable, it would need a demo of its functionality for the advisors.

### Calendar

- Was highly appreciated, only positive feedback.

### General info

- Keeps a lot information easy accessible.
- My portings were especially usable, but not needed for all advisors.

### Proactive customer

- The color coding was clear and helpful for prioritizing.

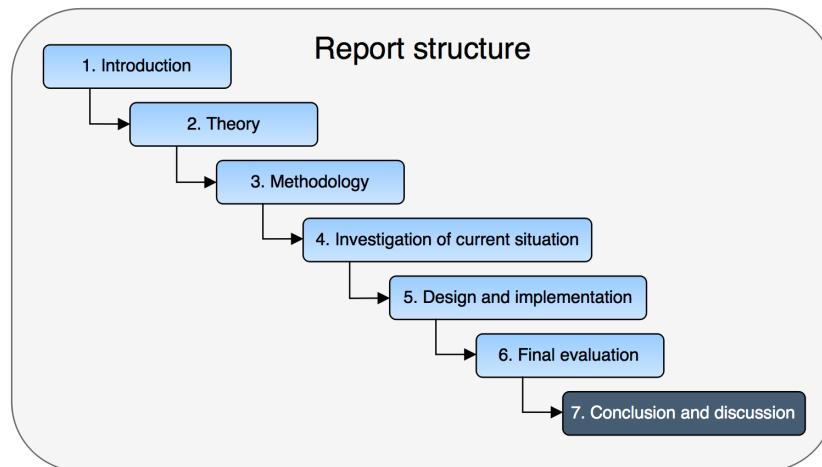
- It must be clearer if the purpose is to identify customer loss rate.

---

## Conclusion and discussion

---

This chapter contains a summary of the research questions, the discussion of the findings in the report and a section that covers future work. In figure 7.1 one can see the part where this chapter belongs in the report structure.



**Figure 7.1:** The report structure with current chapter highlighted.

### 7.1 Conclusions

The thesis has four research questions, stated in section 1.3, that were investigated during the project. These were answered continuously in the report, and done in a linear fashion. This section summarizes the results.

Research question one, that states " *What does literature say is a good process of developing a web tool with focus on interaction design?*", is covered in chapter 2.

It covers general design and code development methodologies that helps creating a web based tool with an emphasis on user experience and interaction design. The literature states that while designing, it is of importance to be well aware of the end user by doing an thorough investigation. Further, developing with an iterative process and keeping a close contact with the intended end user.

The second research question, "*How can work patterns and routines be analyzed to create a better tool?*", is investigated in chapter 4. The outcome from that chapter, resulted in requirements that were obtained in this phase using interviews, observations and quantitative analysis. The requirement identifies the needs in the project and gives a good starting point to implement a new tool in the project.

The third question, "*How could an overview page within Telavox' existing CRM-system, Partner, be implemented?*", is covered in chapter 5. The essence of the chapter, which is also answering the research question, is the process from creating sketches and prototypes in various forms to a fully functional product. The emphasis in the report is on the design aspects, as the aspect of coding is left out for the most part.

The last research question, "*How can an overview page improve work efficiency for customer service at Telavox and enhance Advisors understanding of what can be done?*", is evaluated in chapter 6. The main part of this chapter, that answers this question, is section 6.3, where real users of the dashboard try it out. The outcome of the evaluation was the importance of overview in a tool like the dashboard, and also to be extremely clear about how the user is intended to use the tool. This research question could have been further analyzed by doing a quantitative analysis, similar to the one done in the investigation phase, and compare it to previous data collected of Dashboard 1.0. However, during the evaluation, the general opinion was that the accomplished evaluation was sufficient for answering the question.

## 7.2 Discussion

### 7.2.1 Interview result and user personas

In this thesis two observations were made, three interviews and three user tests. Both the observations and user tests gave a general view of how the situation was or what needed to be changed. However, only three interviews were conducted. The low amount of interviews were primarily done due to the recurring information that was given in the interviews, which would make another interview redundant. This, together with consultation from our supervisor at LTH led us to make the decision to only do three interviews.

In this project, two user personas were developed based on the interviews. It was to create a common picture of whom the project and dashboard that was

designed for. The interviews granted some personal information about age, computer and work experience, and other observations gave a general idea of what a typical advisor could be like.

Even so, to base them of three interviews where the personal depth did not scratch the surface gave no solid foundation to grant the user persona any legitimacy. Even so one primary and one secondary were made in hopes of being useful.

The result of the user personas were never used in any situation. This was due to the ease of getting a hold of the intended users with ease as they work in the same office building. The argument of having the same conceptual model was more easily gathered by walking to an advisor and ask a question then to resonate about a fictional user persona. The project had no emphasis on using user personas, and it might even be better of by not using them and having direct contact with the user instead.

### **7.2.2 Involving the other parties more often**

The project was done without much consultation of Telavox' user experience team. To involve the UX team in an earlier stage, could have helped impact some of the design decisions that were done during the project. One of these were the custom ticket settings view, where a simple conversation could reveal the design flaws that were presented in section 6.2. This, together with the general advice they gave at the end of the project, could have helped improve the result of the final dashboard both in a slightly different design, but also in a time saving aspect.

The main reason they were not involved in the project was mainly due to summer vacation, where the whole team were not present during several weeks of early implementation. After those weeks it became harder to involve the UX team, to the point that they only gave an expert evaluation in the end.

Another party that could have been involved more often is the users, in this thesis the advisors. When developing, there is always a risk of falling into negative work pattern, where the developer does not want feedback of the work done until the product is flawless, which it never will be. In this thesis, the contact with the advisors could have been more frequent, especially in the end of the thesis.

### **7.2.3 "The Partner look"**

The development of a new dashboard had one starting point: the header. As mentioned in section 1.4 the header was kept to not involve the navigational elements and search functionality. This might have set the project in another direction than to evaluate the dashboard with the header in mind.



During the sketching phase some sketches were made without taking these delimitation's in mind. This resulted in a mock up that had a totally new design of the adviser view of Partner. The design was discussed from the point of view of implementing it and how well it would be received. The conclusion of the discussions were that it would require a total redesign of Partner and a much bigger scope to analyze. It was due to these discussions the header delimitation was implemented in to the project.

#### **7.2.4 The concept of proactive watch section**

The customer proactive watch was only developed as a concept. This made the majority of its functions being various placeholder objects in the design. It could affect the result of our user tests, as there was no real data to use during these tests and had to be created. This could give a false view of how the tool could actually work, as it might have overflowed the site, or been worthless in a real life scenario.

The decision to make the feature only a concept was made due to the estimated time it would take to implement it. To make an effective tool that could be used to determine variables, that could be of interest and help Telavox keep a customer in a proactive way, was of a too big of a scope to be handled in this thesis.

However, the importance of such a tool for the advisor's work flow, as evident in chapter 4, made it important to add it into the design. The future development of the tool is mentioned in section 7.2.5.

#### **7.2.5 Future work**

Since there was not enough time to apply a third iteration, it was necessary to set a "feature complete", meaning no more features were to be implemented from that moment on. However, when iteration 2 came to an end, there were still several things to improve on the dashboard. A selection of these ideas and possible improvements are presented below.

##### **Implement a web socket**

To create a better user experience, as much functionality as possible that does not required user interaction should be done automatically. A good example of this is the fact that the advisor must update the ticket views manually. If the advisor forgets to update the view, tickets can be forgotten until several moments later. Equally important, the back end will be burden with unnecessary requests. Instead, this would be a good occasion to implement a web socket, which would push the updates to the client (the advisor's web browser). This feature would also be able to implement on the proactive watch section and similarly the general

info section. This feature would open up new design possibilities such as notifications.

### **Complete back end implementation of proactive watch section**

Since the concept of the proactive watch section was appreciated and also seemed to be a feature that could help the advisors in their proactive work flow, the next step would be to actually implement it. The task would not be extremely complex, but with full functionality it would require several database requests. This would probably make the page slower, which would go against one of the main goals. However, it could be solved by only making the requests once a day, during night time.

### **Improve general info section**

The general info section has been proved useful for several advisors. However, some content was irrelevant for some advisors. A useful feature would be to implement a filter, where the advisor could choose what tabs to view and load into the clients browser. This would also ease the load on the web server. Moreover, it would be valuable to append a statistics tab to satisfy the advisor team leaders. Redesigning the delivery tool to enable more features, e.g. disable columns, AJAX etc. would be useful.

### **A more complete ticket system**

Finally, there are several things to improve in the ticket system section. Some of them are listed below:

- The tree view in the custom tickets settings needs to be completely redesigned to make it more intuitive.
- Avatars instead of the colored circles would make the ticket assignment clearer and recognizable.
- Add more settings where the advisor can choose the amount of recent notes showing. Also the possibility to click a “View 5 more notes” button would be valuable.
- A more complete note submitting functionality, where the advisor has the opportunity to answer the customer and hide tickets until a future date.
- Add possibility to toggle between showing hidden ticket or to not show them.
- Finally, visualize how many open- and hidden tickets there currently are.

### 7.3 Final remarks

This thesis have been a very informative project for the thesis workers. During the project, many insights have been discovered, both by the thesis worker themselves, but also from friends, family, supervisors and especially from the all the colleges at Telavox. Two of the greatest findings during the project was the importance of thorough investigation of the domain and to keep close contact with the customer. The knowledge will, without doubt, be used in future work.

---

## References

---

- [1] Telavox AB. About telavox. <https://www.telavox.com/en/om/telavox/>. Accessed: 2016-06-10.
- [2] A. Lantz. *Intervjumetodik*. Studentlitteratur AB, 3 edition, 2013.
- [3] M. Höst, B. Regnell, and P. Runeson. *Att genomföra examensarbete*. Studentlitteratur AB, 1:6 edition, 2006.
- [4] M. Arvola. *Interaktionsdesign och UX*. Studentlitteratur AB, 2015.
- [5] D. Norman. *The Design of Everyday Things*. Basic Books, 2013.
- [6] S. Lauesen. *Software Requirements*. Addison-Wesley Educational Publishers Inc, 2002.
- [7] R. Sefelin, M. Tscheligi, and V. Giller. Paper prototyping - what is it good for? a comparison of paper- and computer-based low-fidelity prototyping. *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, pages 778–779, 2003.
- [8] M. Rettig. Prototyping for tiny fingers. *Communications of the ACM*, 37 Issue 4:21–27, 1994.
- [9] J. Gould and C. Lewis. Design for usability: Key principles and what designers think. *Communications of the ACM*, Vol. 28, 1985.
- [10] B. Bruegge and A.H. Dutoit. *Object-Oriented Software Engineering Using UML, Pattern, and Java*. Pearson Education, Inc., 3rd edition, 2010.
- [11] S. Krug. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders, 2014.
- [12] Bootstrap. <http://getbootstrap.com/>. Accessed: 2016-06-30.
- [13] J. Porter. Testing the three-click rule. [https://articles.uie.com/three\\_click\\_rule/](https://articles.uie.com/three_click_rule/), 2003. Accessed: 2016-06-15.

- 
- [14] N.H. Mackworth. Visual noise causes tunnel vision. *Psychonomic Science*, pages 67–68, 1965.
- [15] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mellor, K. Schwaber, J Sutherland, and D. Thomas. Agile manifesto. <http://www.agilemanifesto.org/>, 2001. Accessed: 2016-06-20.
- [16] J. Shore and S. Warden. *The Art of Agile Development*. O'Reilly Media, Inc., 2008.
- [17] Chromatic. *Extreme programming pocket guide*. O'Reilly Media, 2003.
- [18] B. Appleton, S.P. Berczuk, R. Cabrera, and R. Orenstein. Streamed lines: Branching patterns for parallel software development. *PLoP '98 conference*, 1998. [http://www.hillside.net/plop/plop98/final\\_submissions/P37.pdf](http://www.hillside.net/plop/plop98/final_submissions/P37.pdf).
- [19] W. Babich. *Software Configuration Management: Coordination for Team Productivity*. Addison-Wesley Pub, 1986.
- [20] J. Nielsen and B. Curtis. Applying discount usability engineering. *Journal IEEE Software*, 12, 1995.
- [21] D. Kane. Finding a place for discount usability engineering in agile development: Throwing down the gauntlet. *ADC '03 Proceedings of the Conference on Agile Development*, 2003.
- [22] J. Nielsen. Why you only need to test with 5 users. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>, 2000. Accessed: 2016-06-17.

## Interview plan(in Swedish)

---

### Inledning

Presentation: Vilka vi är, att vi exjobbar, bygger en ny dashboard för Partner.  
Syfte:

- Långsiktigt: Intervjun syftar till att vi ska förstå vilka verktyg som används i Partner idag för att sedan kunna ta fram vad som ska finnas på den nya dashboarden. I sin tur ska även dessa verktyg ge en bra användarupplevelse.

Intervjun kommer pågå i ungefär 45 minuter och vara relativt öppen, vilket innebär att frågorna kan tyckas vara något ospecifika. Detta är meningen och du svarar enligt bästa förmåga. Ta gärna med och beskriv känslor som uppstår . Vi kommer att under intervjun spela in, detta är mest för att vi vill kunna lyssna i efterhand. Intervjun kommer vara anonym, i form av att du får ett tilldelat pseudonym i rapporten.

Fungerar detta bra?

Vi kommer börja prata lite om dig som person och din roll på Telavox, sedan fortsätter vi att prata lite om Partner i allmänhet och sist går vi in på detaljer om dashboarden.

### Frågefas

Om personen:

- Berätta om dig själv och din roll på Telavox.
- Hur länge har du jobbat som advisor?
- Hur ser en vanlig dag ut?

Partner:

- Hur hjälper Partner dig i ditt arbete?
- Vad för verktyg använder du mycket under din arbetsdag?
- När känner du att arbetet blir repetition och ineffektivt, var kan man spara tid?

Dashboard:

- Hur ser dashboarden ut idag? (Utan att kolla)
- Hur använder du dashboarden idag?
- Kan du beskriva genom att använda datorn ditt workflow från att en kund ringer in och vill ha hjälp med något?
- Finns det någon sida där man ofta utgår ifrån i sitt workflow?
- Finns något att bevara och förstärka?
- Vad tycker du om en dashboard?
- Hur skulle en ny dashboard kunna hjälpa dig i ditt arbete?
- Vad är målet för en dashboard enligt dig?
- Vad är dina förväntningar på en dashboard?
- På vilket sätt kan detta hjälpa dig i ditt arbete?
- Vad är en relevant dashboard för dig?
- Kan du berätta om något tillfälle du har använt den nuvarande dashboarden?
- Kan du berätta om ett tillfälle då du skulle vilja använda den, men du har fått göra på ett annat sätt?
- Kan du förklara vad som skulle kunna få dig att få en god uppfattning vad du ska göra under dagen när du kommer till jobbet på morgonen?

Följdfrågor, tillämpas när det passar in:

- Hur menar du när du säger..?
- Vad gör du nu?
- Hur tänker du nu?
- Vad lägger du särskilt märke till i detta läge?
- Finns det några tumregler för hur man ska göra i detta skede?
- Vilken information använder du här, hur får du tag i den, och vad gör du med den?
- Under vilka förutsättningar fungerar det på det viset?
- Varför är det viktigt med..?

## **Avslutning**

Nu är vi klara, jag tror verkligen vi kommer kunna använda denna data i det fortsatta arbetet. Har du några andra tillägg eller frågor till oss? Vi finns tillgängliga på kontoret eller via mejl om det är några frågor eller likande.

Tack



## User personas

---

### **B.1 Primary user persona: Anna Almér**

Anna has worked at Telavox for 12 months as an advisor. She is 27 years of age and lives in the outskirts of Lund, taking the train every morning to work. She is an active athlete, spending weekends and nights training as she is a former youth high jump district champion, but now just remains active to stay fit and also coach younger athletes. She lives together with her high school sweetheart whom she also owns a dog together with.

She has previous experience in customer relations at another telecom company, Telenor, previous to working at Telavox, and is used to handling customer problems as they call in. She worked there for two years, but felt it was time to change due to the stale work environment.

During her time at Telenor she got accustomed to systems with large faults and tedious work processes. This was a change when coming to Telavox due to its relative ease and “user friendly” environment and she is somewhat hardened from the previous work environment. Even if she feels that the work environment is better at Telavox, she feels that some processes are made to difficult and could be streamlined.

### **B.2 Secondary user persona: Simon Hammarström**

Simon is a team leader at Telavox and has worked there for three years. Simon is 29 years old and lives in the heart of Malmö. Simon is a great fan of literature and spends time both behind books reading them, but also wants to one day write a novel, even though time right now is missing due to work.

His role at Telavox is split between being a part time advisor and team leader. Spending 50% of his time as an advisor, he takes calls solving ongoing problems

but also try to proactively solve problems that may arise in the future.

As he is an experienced employee at Telavox and team leader he often handles questions from other colleagues that is of various form, from invoice questions to more technical or system specific.

He is in charge of a 6 persons team that handles companies with 11-25 users, where every member of the team has around 160 companies to handle. Simon has about 55 of these himself due to his administrative role as team leader.

His role involves him having the responsibility of distributing companies to other team members, having overview of workload and have meetings with his and other teams. This can lead to him having other members taking care of his customers sometimes when Simon is not available. For Simon it is extra important to have an overview and to get a quick and easy hold of what is currently going on in the team.

## Sketch illustrations

---

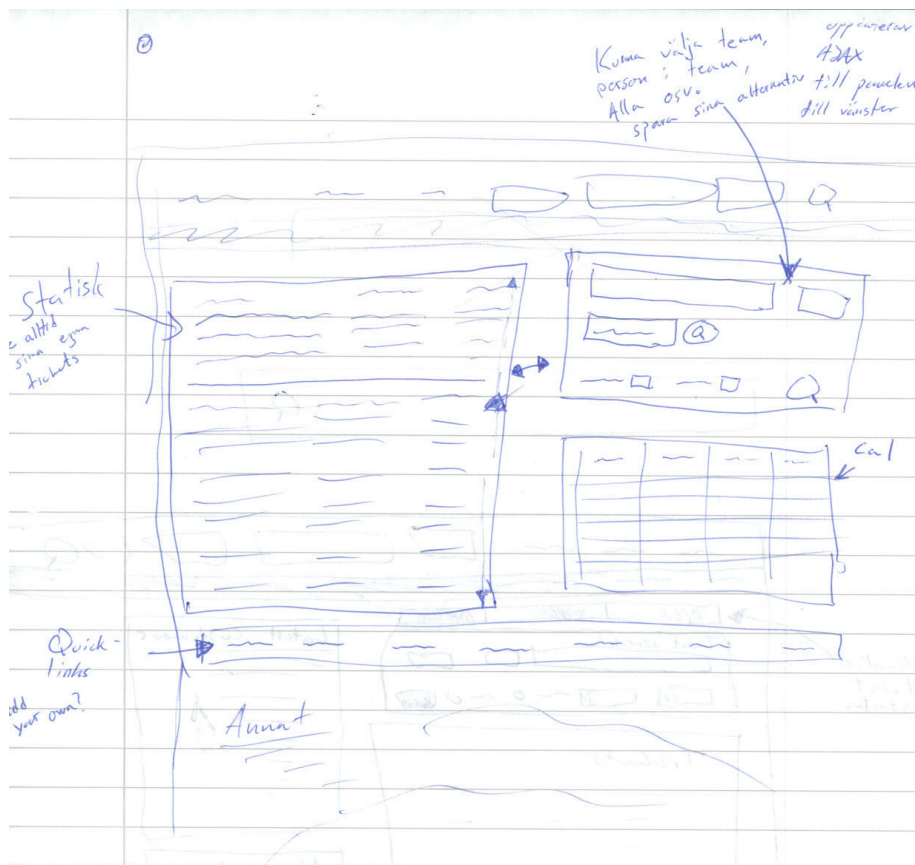
### C.1 Paper sketches

All sketches during this project were done based on the information gathered in the requirement tables 4.4 and 4.5 contained in the report. The sketches were done in the processing phase, described in section 2.2.2. The main focus was to get a general goal to aim towards, and a lot of the sketches look similar due to the nature of the requested features in the two requirement tables.

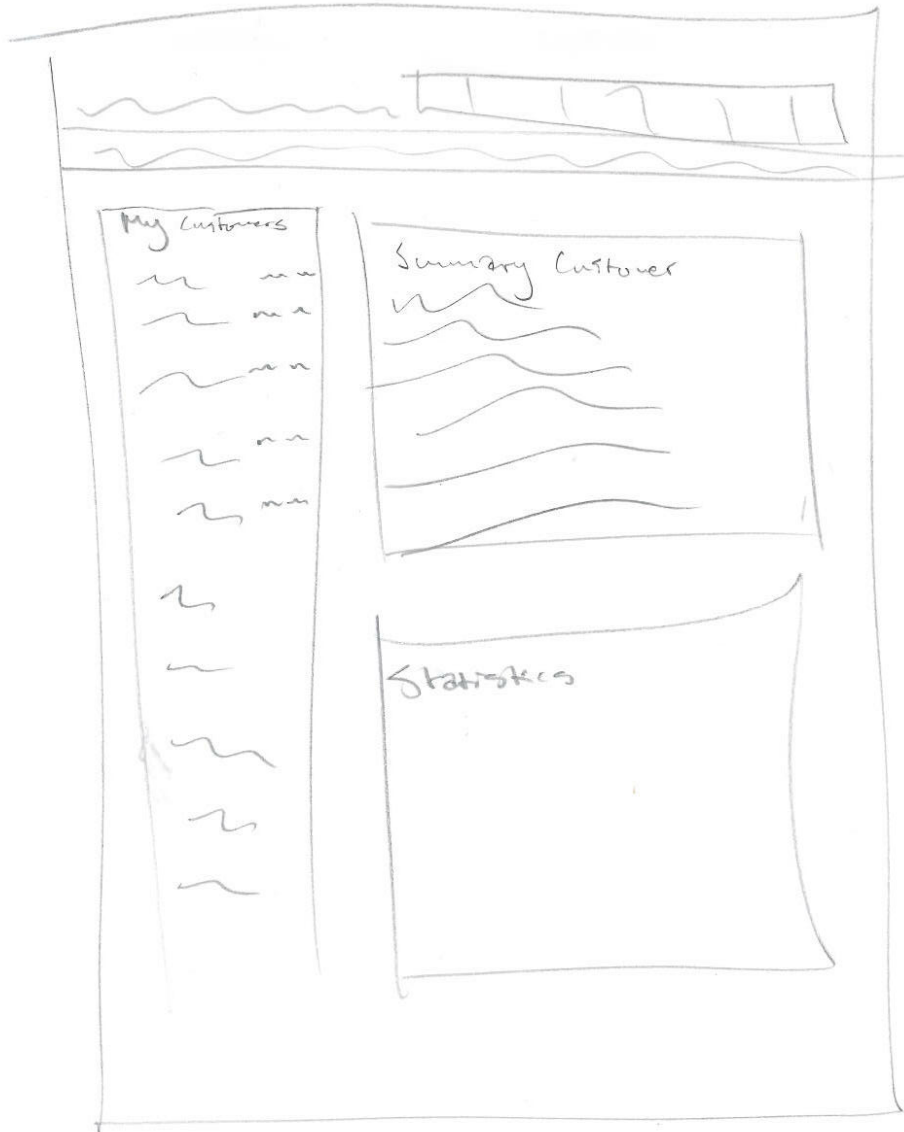
Sketch C.1, C.3, C.4, C.6, C.7, C.10 and C.12 all share common traits as a ticket handling system, card based presentation style and other various functions to display information.

Interesting sketches to note is for example C.11, referred to as the Pokéball-sketch. Every ball represents a customer, and depending on size the various urgency to handle the different problems they could have, the bigger the ball became.

Other sketches such as C.2 presented summaries of customers with various information and a list that could rank the importance of the customer task at hand. Sketches C.5, C.8 and C.9 all contain one of the “vital” elements that could be used on the dashboard, with just a focus on that particular feature, all to keep the features simple and clear.



**Figure C.1:** The first sketch, containing a ticket handling system, search functionality, calendar and quick links.



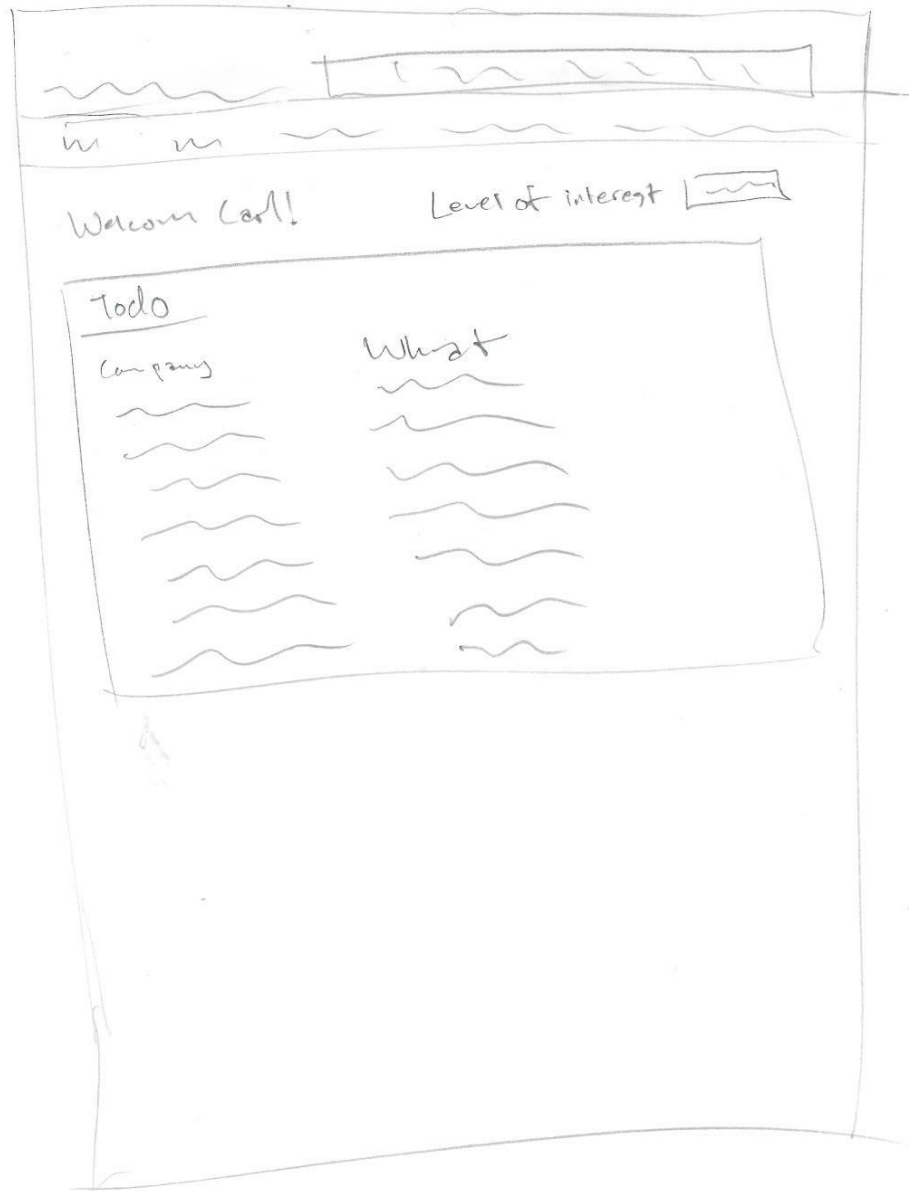
**Figure C.2:** The second sketch, containing a list of all the logged in users customers, summary of them and a statistics view.



**Figure C.3:** The third sketch, containing a small customer list, statistics view, calendar and a ticket view.



**Figure C.4:** The fourth sketch, containing a ticket view, calendar, upcoming portings, statistics, campaign- and news view.



**Figure C.5:** The fifth sketch, containing a broad to do list for the logged in user.





**Figure C.6:** The sixth sketch, containing various sections that can be open and closed.



**Figure C.7:** The seventh sketch, containing a ticket view, a customer list and a tab view.

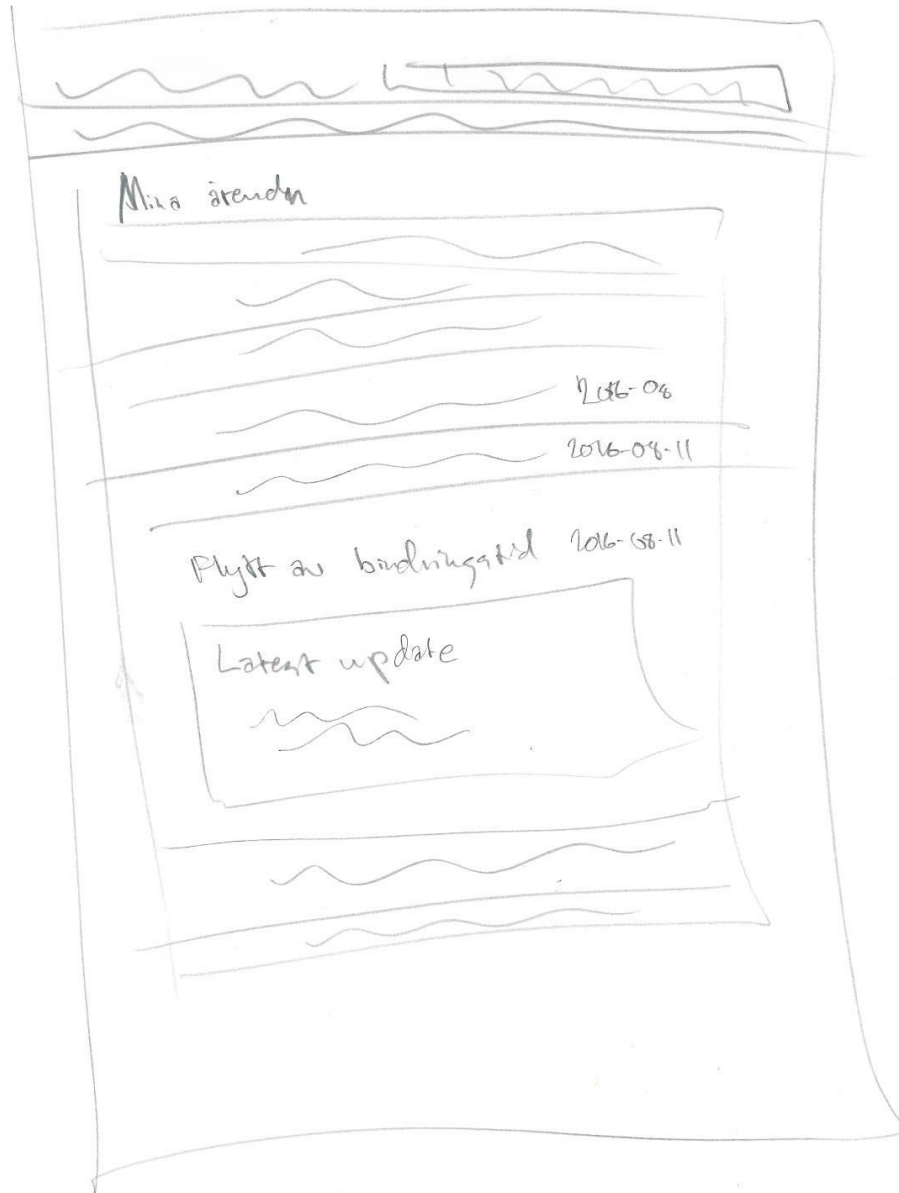
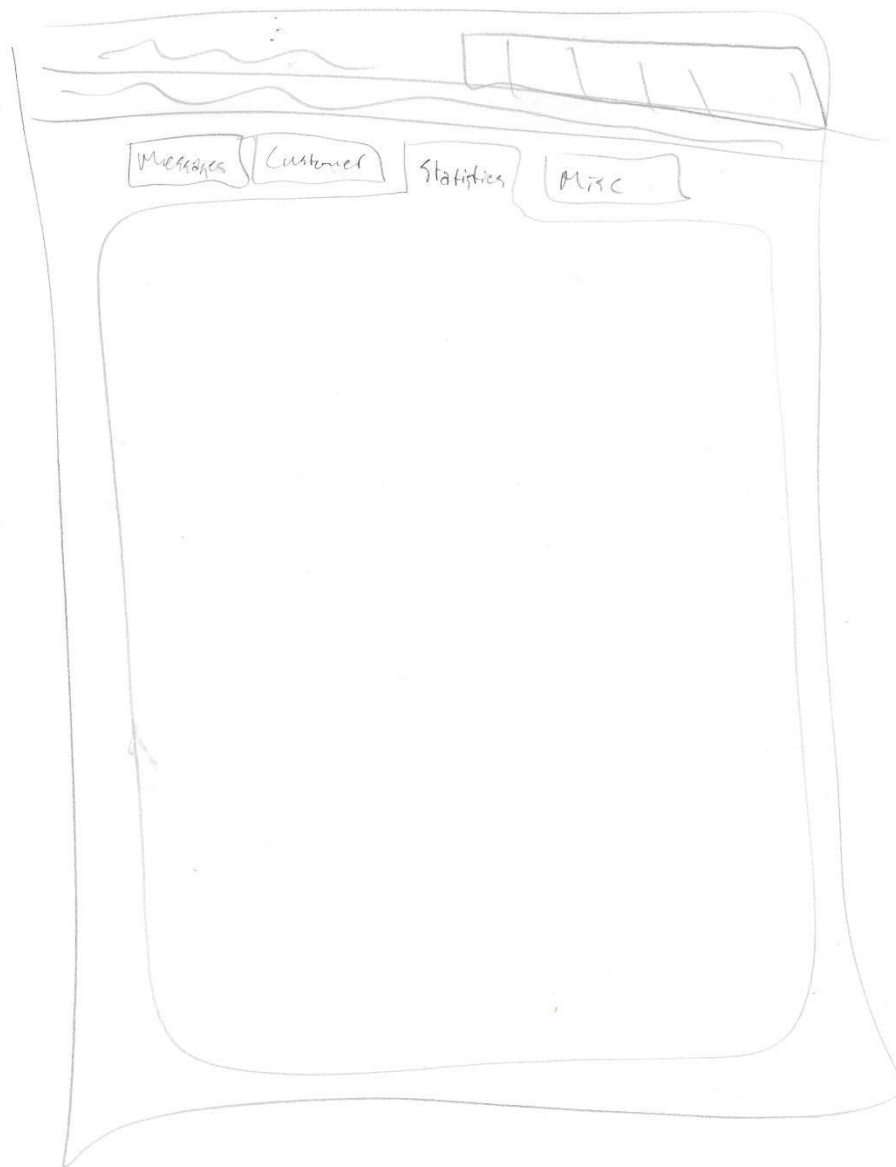
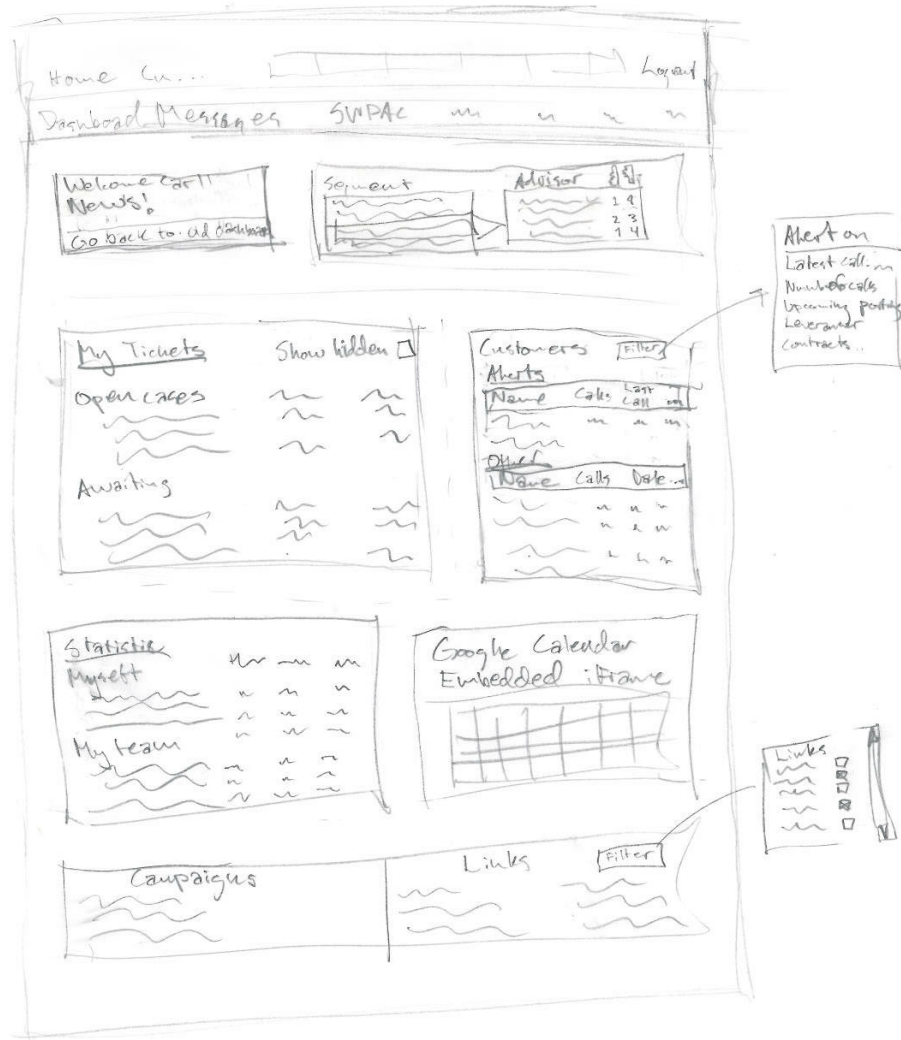


Figure C.8: The eighth sketch, containing a big ticket view only.



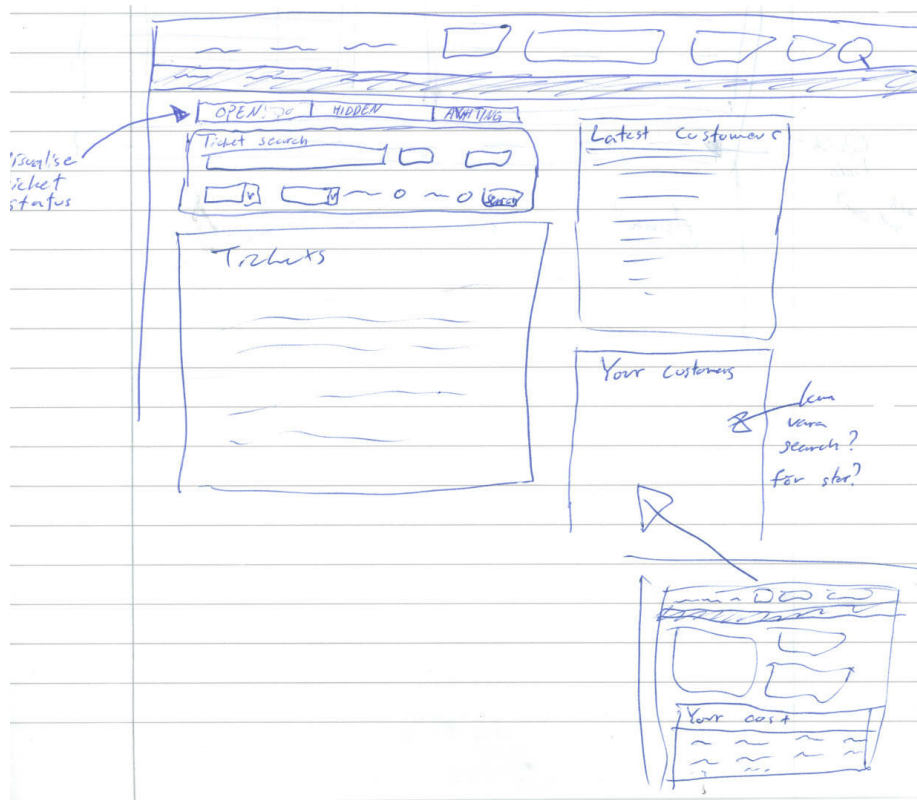
**Figure C.9:** The ninth sketch, containing a big tab view with various functionality under different tabs.



**Figure C.10:** The tenth sketch, containing a summary of various other users, a ticket view, customer view, statistics, calendar, campaign info and links.



**Figure C.11:** The eleventh sketch, containing customer balls that vary in size depending on importance.



**Figure C.12:** The twelfth sketch, containing a ticket view and overview, and two customer views.

## C.2 Whiteboard sketches

The first sketch, C.13, had four different main features placed in a two by two grid. Urgent information as ticket info and a search bar were paired together to create a mapping. To create a visual overview a pie chart was added to visualize the user's ticket statuses. A tab field that could handle various information without cramming the page was added to reduce noise and add a relevance to the information one chooses to display. Also a customer overview field was added to create a quick way to view customers. The sketch also highlights the question of grouping the parts relevant to the logged in user, and the general information fields.

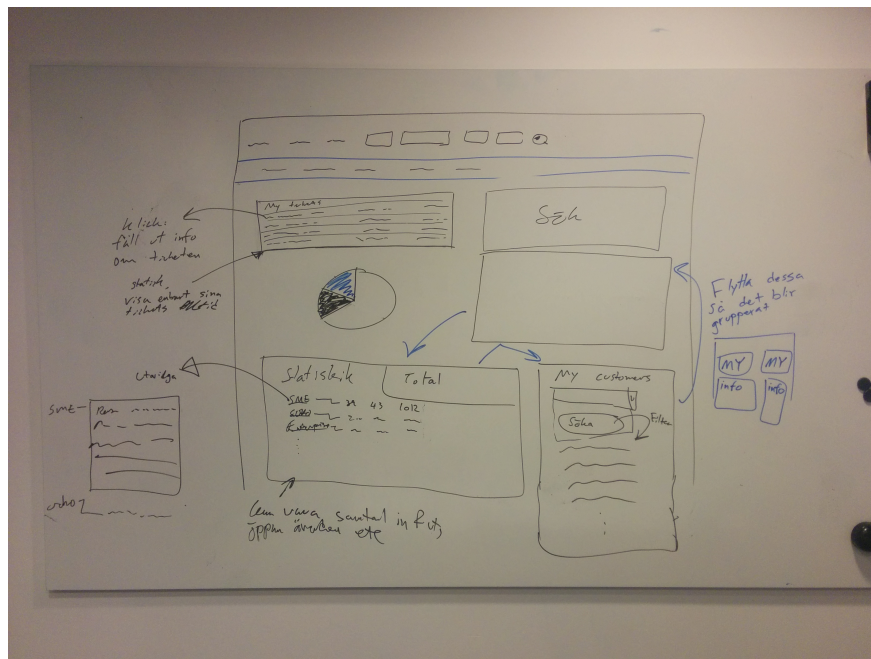
Sketch C.14 adapts the ticket view system, with an added feature to distinct a user's own assigned tickets and other custom searched tickets. A view with statistics is present, and on the bottom is an automatic proactive service that pushes notes to the advisor with information of ways to help customers proactively. This sketch is slightly more focused due to less features implemented, and more space devoted to the proactive tool to encourage the way of working proactively.

The final sketch, C.15, once again has a ticket field in the top left corner. The proactive feature from sketch C.14 was added in a smaller view to the right. This grouped the two ticket work related tools, and left the other two planning and information tools below. In the bottom left a calendar was added, and beside it a tab view to present different kind of information was added.

One common denominator in all the whiteboard sketches is both the feature and placement of the ticket system. Due to it being currently positioned at the top left corner in the Dashboard 1.0 and the importance it was proven to have when collecting information of what to develop it seemed important to keep it in the same area to take advantage of the current flow the advisors have.

All sketches were also narrowed down to be able to implement in a screen size of a normal full sized screen (1920x1080 pixels).





**Figure C.13:** The first whiteboard sketch, containing a ticket view and a search functionality, a tab view with statistics and a customer view.

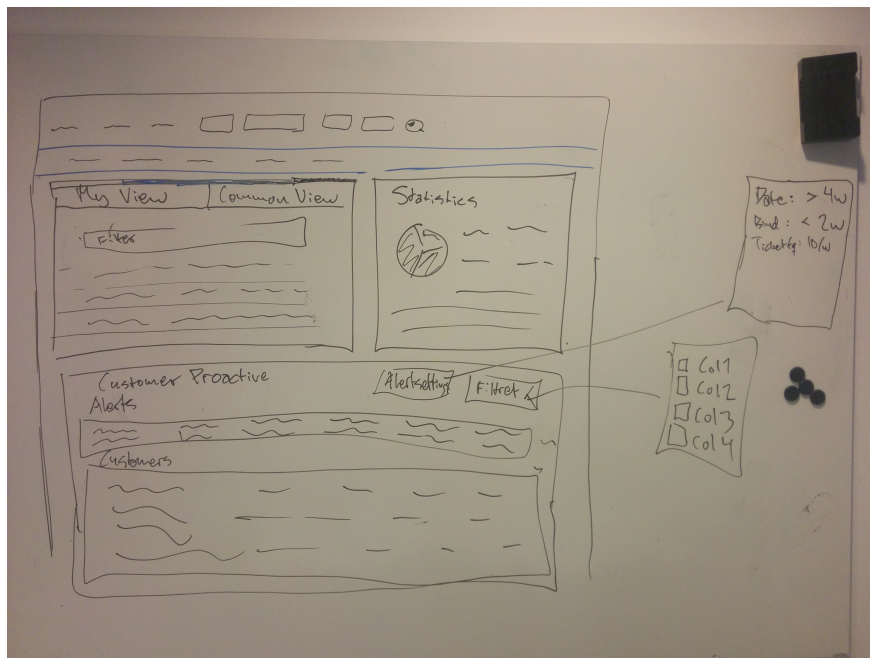
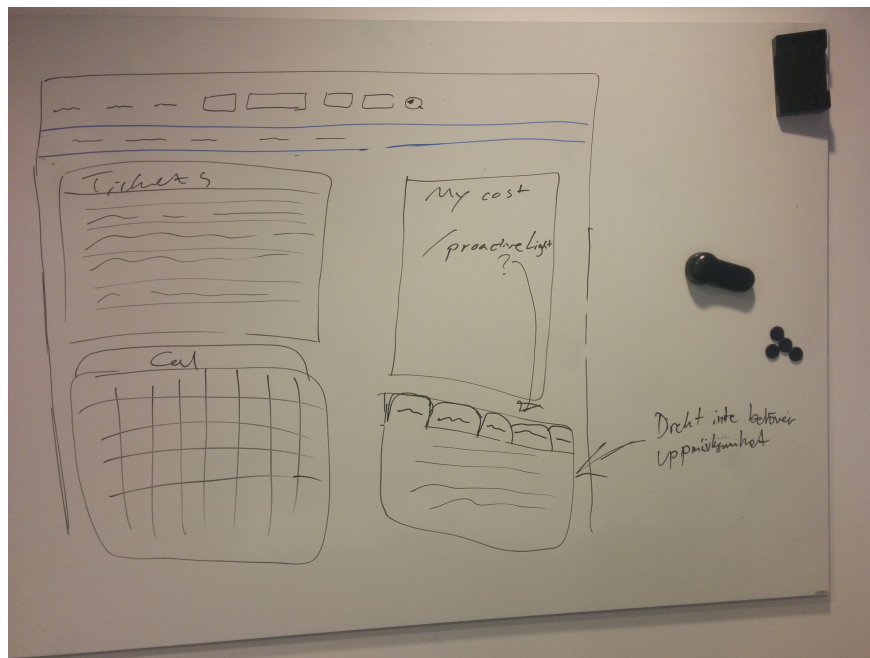


Figure C.14: The second whiteboard sketch, containing a tabbed ticket view, a statistics view and a customer proactive view.



**Figure C.15:** The third whiteboard sketch, containing a ticket view, a customer list, a calendar and a tabbed view with various information.