Designing and Implementing an Application for Data-driven Decision-making

Oskar Jönsson

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2022

MASTER THESIS

Grade

Designing and Implementing an Application for Data-driven Decision-making

Oskar Jönsson

LUND
UNIVERSITY

Designing and Implementing an Application for Data-driven Decision-making

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

Subject: Interaction Design MAMM01
Division: Ergonomics and Aerosol Technology
Supervisor: Joakim Eriksson, joakim.eriksson@design.lth.se Rickard Nygren, rickard.nygren@gade.com Arvid Pilhall, arvid.pilhall@gade.com
Co-supervisor:
Examiner: Günter Alce, gunter.alce@design.lth.se

Abstract

With data becoming more and more abundant, companies will want to use this data in meaningful ways to gain competitive advantages on the market. One way of utilizing the data is to make data-driven decisions, which is often called data-driven decision-making. The goal of this master's thesis is to develop and design an application for the company Grade AB to alleviate data-driven decisions made in the company. This by supplying data from Grade's customer directly to the users needing it, with the help of the application. The thesis focuses on using an interaction design process along with having a user centered design in mind when both developing and designing the application. The result of the thesis is a finished application that the employees can start using internally.

Keywords: Data-Driven Decision-Making, Interaction Design, Prototyping, User-Centered Design, React, ASP.NET

Sammanfattning

Eftersom information blir mer och mer tillgänglig börjar företag idag leta efter potentiella sätt att använda informationen för att blir mer attraktiva på marknaden. Ett sätt att utnyttja informationen på är att stödja datadrivna beslut. Målet med detta examensarbetet är att utveckla och designa en applikation som underlättar datadrivna beslut för företaget Grade AB. Detta genom att erbjuda behövande användare med information på deras kunder via applikationen. Examensarbetet fokuserar på att använda en interaktionsdesigns process och ha användarcentrerad design i åtanke både vid design och implementering av applikationen. Resultatet av examensarbetet är en fullständig och användbar applikation som anställda på Grade kan börja använda.

Nyckelord: Data-drivna beslut, Interaktions design, Prototyp, Användarcentrerad design, React, ASP.NET

Acknowledgements

Firstly, an immense thank you to Grade for the cooperation on the master's thesis. You have been nothing but welcoming and I am very thankful for the opportunity that you have provided me with. I am also very thankful for everyone who has been involved in the project. A special thank you to all the supervisors of the project, Rickard Nygren, Arvid Pilhall, and Joakim Eriksson, who have helped guide me through the project. A final thank you to all participants who have volunteered to help me through the thesis.

Lund, June 2022

Oskar Jönsson

Table of contents

List of acronyms and abbreviations

Low-Fi - Low Fidelity

Mid-Fi - Mid Fidelity

Hi-Fi - High Fidelity

SUS - System Usability Scale

API - Application Programming Interface

REQ - Requirement

RQ - Research Question

WCAG - Web Content Accessibility Guidelines

1 Introduction

This chapter includes the background of the thesis to give the reader an understanding of what gave inspiration and why the thesis was carried out. The chapter also presents related work, purpose and limitations that was tackled in the thesis along with a description of Grade AB where the thesis were carried out.

1.1 Background

With data becoming more abundant for each passing year, companies want to find ways of using the data to increase their competitiveness on the market [1]. One way of capitalizing on data is to support the company's decision making process by providing information to alleviate the decision maker. This is called data driven decision making which is when the decision maker uses metrics and data to guide the decision making process. The amount of collected data has never been higher and it is forecasted to increase even more in the coming years [1], with all this data there is a need to take into consideration in why, what and how it is presented. This master's thesis was a collaboration with the company Grade AB which distributes an e-learning platform for its customers. Grade AB wants to use the data they have collected on their customers to enhance and support decisions taken in their company. To achieve this, Grade proposed the task of designing and implementing a web based application that presents this information to the employees and stakeholders in the company along with a supporting infrastructure.

1.2 Grade AB

Grade AB is a Lund based company that provides a platform which combines e-learning, talent management and pulse checking, called GRADE [2]. The platform is built to support more effective and agile human resources processes and tries to create a cycle of competence, learning and engagement. The GRADE platform is split into three modules ENGAGE, LEARNING and TALENT which works seamlessly with each other. ENGAGE is a pulse checking tool where you can measure the

engagement of the workplace by sending out surveys with relevant topics. Management can then view the result from these surveys in intuitive dashboards. TALENT is a talent management tool for keeping track and acquiring talent in the company by supplying functionality such as staff appraisal and competence management. The goal of the module is to have right person with the right competence at the right time. The LEARNING module is a learning management system with the goal to increase competence through the organisation. This by supplying the organisation with activities such as online courses.

1.3 Related Work

There are a lack of studies related to developing solutions for data driven decision making, particularly following user centered design. However a large part of the project is related to the work done by Petersson and Holm and their master thesis on Business Intelligence (BI) [3]. In their thesis they developed a solution for gathering data from the large quantity of customer databases that Grade provides and stores it into a single database or so called data warehouse with different applicable dimensions. With this work they also created a temporary web based interface for displaying the data gathered in the warehouse. The data warehouse solution was a big part of the implementation phase of the project and without the underlying architecture that this provided the project would not have been possible in the allocated time period.

1.4 Purpose & Research Questions

The purpose of this project is to build a complete web based solution for displaying data to alleviate data driven decision making using the interaction design process that can be used by internal employees in Grade. Because the aim is to build a working solution the scope of the project will include activities from designing and implementing to deployment. With this purpose in mind four goals and three research questions were established.

Goals

- Utilize the interaction design process to create the solution.
- Utilize the concepts of user centered design to create the solution.
- Create a High-Fidelity prototype showing the design of solution.
- Implement the requirements to make a functioning application.

Research Questions

RQ1: How can the interaction design process be used for creating a solution for Grade?

RQ2: How can the concepts of user centered design be used to meet the users expectations at Grade?

RQ3: How can the implementation of the final design be created?

1.5 Scope & Limitations

The main limitation of the project was the allocated time available. Because the project were done as a master thesis it was limited to a period of 20 weeks consisting of 40 hours of work. Because the primary purpose of the thesis is to deliver an actual working solution, every step from user research to implementation had to be done. This in addition to the time constraint meant that there were a time limit on every step of the process and therefore some steps had to be rushed even though they were important. The project was also limited by the quantity of data Grade could provide on their customers and because of the time constrains acquiring new sources of data was hampered. Along with the data the pre built infrastructure from the previous master thesis also served as a limitation, with the solution being bounded by what the infrastructure could handle.

2 Theoretical & Technical Background

This chapter introduces the concepts of user centered design and the interaction design process. Then several other concepts and techniques are presented in the areas of data gathering, idea generation and prototyping, with the chapter ending with a technical background for the thesis.

2.1 User Centered Design

Gould and Lewis [4] presented three design principles that according to Mao et al. [5] are today accepted as the basis for user centered design. The principles are the following:

- Early focus on users and tasks.

As suggested by its wording this principle is supposed to help the designer get an early understanding of who the users are and what task they are trying to accomplish. This is done by studying multiple aspects of the users such as cognitive and behavioural characteristics with the help of observations of the user.

- Empirical measurement

The second principle aims to help the designer get measurements of the users performance with the product as early in the development process as possible. This by evaluating the performance of the intended users on simulated experiences such as prototypes and scenarios.

- Iterative design

The final principle is meant to emphasize that issues found when conducting the user test needs to be fixed and then tested again. This makes the process iterative and it follows cycles of design, evaluate and redesign. This iterative process will be described in the next section.

2.2 Interaction Design Process

The interaction design process contains of the four essential activities [6]:

1. Establish requirements
2. Design alternatives
3. Prototyping
4. Evaluating

Interaction design is a iterative process and the activities should be repeated through the entire design process. Information should also propagate through the activities where one activity build upon the work of the previous [6]. To be able to design a product that supports its end users the designers need to know the requirements of the users. This is the first of the four essential activities, establish requirements. The method of establishing requirements is simply finding out what the end users need to succeed in the product that is being developed. Instead of guessing these needs designers can conduct data- gathering and analysis activities [6]. When requirements are set the actual work of designing begins. The activity designing alternatives is coming up with ideas that satisfies the specified requirements. According to [6] this can be split into two sub concepts: conceptual design and physical design. Conceptual design is the process to the create a conceptual model for the product and physical design in the graphical design of the product including colors, themes etc. [6]. To be able to test and evaluate the design, users need something actual to interact with which is where the prototyping activity comes in [6]. Prototyping is the act of creating a interactive prototype of the design created in the last activity. The prototype does not need to be in the same domain as the finished product which means that a paper based prototype can be used for digital products like software. The last activity evaluating is the process of deciding the quality of the design. This could be for example the usability and acceptability of the design produced [6]. To determine the quality there are several activities a designer can perform such as heuristic evaluations or usability testing.

2.3 Design Principles

2.3.1 Norman's Principles

To be able to conform with a user centered design, Norman provides a set of principles the designer can use to aid their decisions making [7].

Affordance

Norman describes the term affordance as the mapping or relationship between a entity and a person [7]. The relationship gives the person a clue of what possible actions the entity can process. For example a button gives the interactor the clue that it can be pressed.

Signifiers

As described before Norman defines affordances as the relation between a entity and a person. Signifiers is a way of enhancing the affordances of an entity by providing perceivable indicators. Examples of signifiers could be arrows, labels or icons because they guide the interactor to the right affordance.

Mapping

Mapping is the relation between inputs and outputs or as Norman describes it, controls and displays [7]. The more natural the mapping is the easier it is for the interactor to figure out how they should be used [7]. An example of a natural mapping is the relation between a desktop mouse and its virtual counterpart, when the mouse is moved horizontally on the mouse pad the virtual mouse is moved in the same direction with the same speed. If the interactor speeds up the movement of the mouse or changes directions, the virtual mouse will follow.

Feedback

Norman defines the principle of feedback as the information the interactor gains from taking an action [7]. Feedback is essential for performing task because without it the interactor has no way of knowing if there were a affect to the action. Sounds or visual cues are one of many ways of presenting the interactor with feedback, for example a doorbell provides a signal when pressed.

Constraints

Constraints is the principle of limiting the different ways the interactor can interact with an entity. In the example with the desktop mouse the virtual mouse is constrained to the frame of the computer screen. Norman declares the different forms of constrains as physical, logical, cultural and semantic [7].

2.3.2 Web Content Accessibility Guidelines

Web Content Accessibility Guidelines (WCAG) is a set of guidelines published by the Web Accessibility Initiative [8]. These guidelines aim to provide web developers with the knowledge of designing inclusive websites [6]. WCAG embraces four principles of inclusion, the website must be operable, perceivable, understandable,

and robust [8].

2.4 Data Gathering & Analysis

2.4.1 Quantitative & Qualitative Data

There are two different categories which research data can be divided into, quantitative or qualitative. Quantitative data is data that either is made up by numbers or can be represented with numbers and can often be used together with statistical analysis [6]. For example gathered data of population or income is quantitative data. Qualitative data is data that cant be easily be represented with numbers [6]. For example quotes, thoughts and preferences is all examples of qualitative data. It is important to distinct that a single gathering method does not need to result with data in one of the corresponding categories but can yield both [6].

2.4.2 Interviews

An interview is one of the most traditional ways of extracting requirements from stakeholders in the product [9]. A interview in principle is very simple, the interviewer ask questions to the interviewee regarding specifications and requirements of the product and the interviewee answers them [9]. According to [10] interviews falls into one of three categories: unstructured, structured and semistructured. Unstructured interviews is conducted without predefined questions and it is often used to explore the interviewee within a set topic and is often suited in the early stage of the life cycle [6]. Unstructured interviews often use open questions where the interviewee can provide a detailed answer. A structured interview is the opposite where the interviewer uses predefined closed questions where the answers often fall into a defined set. This approach gives a more quantitative result [6]. Semi-structured interviews is a middle ground between unstructured and structured and it combines both open and closed questions [6].

2.4.3 Observations

An observations is another data gathering technique with a high involvement of the user [9]. With certain techniques such as interview it can be difficult for the interviewee to provide a proper explanation of why they perform certain tasks. By observing the user perform the task the observer can explicitly see the user perform the task and therefore get more information than the user can provide in for example a interview [6]. Observations can be used through the whole design process to get data on the

users context, and tasks or to help evaluate prototypes [6]. The observations can be performed either in a controlled environment or directly in the field. A observation in the field can help find clues regarding the context of the user and certain nuances the user faces in their environment and the observer gets the whole perspective of how the user performs its tasks [6]. A controlled environment observation bring more formality to the observation and improves the repeatability of the observation because it has more structure and controlled variables.

2.4.4 Think aloud

Observations only show the actions of the user when performing a task but not their thought process behind why they take those actions. Sometimes the designer would like to have information surrounding that process but at the meantime observations should not be interrupted by the observer with questions because it makes the observation intrusive [6]. To alleviate this there is a method called the think-aloud technique that the observer can instruct the user about. The think-aloud technique as it name suggest encourage the user to speak their thoughts to the observer [6]. For example the user tells the observer that he presses the button as the observer instructed. A problem with the think-aloud technique is that it can eliminate certain problems the user might have faced because the user reminds themselves what they were doing when they are speaking their thoughts [11].

2.4.5 Affinity Diagram

Certain data gathering methods like interviews can produce a large quantity of unstructured data. Affinity diagrams are a good method to organize and bring hierarchy to this data [12]. The diagrams helps the designers see the scope of the project because it gathers all problems and worries with the project in one place. Affinity diagrams are often built using a bottom up process which means that the categories or hierarchy are not predefined and it naturally evolve from the gathered data [6, 12].

2.4.6 Personas

When designing interfaces for users people often fall into the trap of the *elastic user* were the term refers to all users of the application as the singular user [13]. This leads to this user's goals and motivations constantly changing to fit the variety of multiple users, to alleviate the problems Cooper proposes the use of a method called personas [13]. A persona or multiple personas is a abstract generalisation of the end users of the product and it often contains a detailed description of the user and its goals and motivations. According to [9] personas are a great way of defining users

which makes it easier to produce a product matching the users characteristics. Even though personas are fictitious descriptions of users they are still created from the real users. Therefore it is important to gather information from real users to make generalisations to base these personas on [13].

2.4.7 Heuristic Evaluation

A heuristic evaluation is an informal usability evaluation method proposed by Nielsen et al [14]. The method proposes a set of heuristic or usability principles which experts can use to evaluate the design of an interface without involving users [6, 14]. The ten heuristics proposed by Nielsen et al [15] is:

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

2.4.8 System Usability Scale

System Usability Scale (SUS) is a simple scale for gathering subjective assessments on the usability of a system [16]. SUS is built up with ten questions that can be answered giving a ranking from the Likert scale, which gives five options with the extremes being strongly disagree and strongly agree [16]. The questions are predefined to a format which makes it easy to compare between different systems. Each item of the SUS yields a score between 0 to 4 but by multiplying it with 2.5 a range between 0 to 10 is obtained. Combining the scores from the 10 questions gives an overall score range of 0 to 100 [16]. This score can then be evaluated using a curved grading scale proposed by Sauro and James [17] where the score can be mapped to percentiles. These mappings gives the SUS score of 68 the percentile rank 50%

indicating an average score and therefore scoring higher than 68 indicates a higher than average usability. Sauro and James also translates these percentile rankings to grades between A to F [17] with the score of 65 to 71 leading to the grade C.

2.5 Conceptual Design

2.5.1 Scenarios

Scenarios are fictional stories which aim to visualise significant activities from the end product early on in the development cycle [9, 18]. Scenarios has the ability to make designers envision how the use activities are performed before they are created which creates support for discussion and reasoning in the design process [18]. Scenarios are merely stories but they can be used as a base for other forms of prototypes such as storyboards [18].

2.5.2 Project Breakdown Structure

A Project Breakdown Structure (PBS) is a a hierarchical diagram of items that the project will create or deliver [19]. The PBS is meant to provide a visual overview of the whole system and give the participants confidence of what is required to finish the project [20].

2.5.3 User Flows

User flows are diagrams that describes the specified path the user would take when performing a task in the product [21]. By specifying user flows designers put more focus on providing the user what they want from the product and it makes it easier to prioritise user experience over graphical design. The diagrams also helps the designer spot issues where the flow is more complicated than it needs to be or come up with alternative paths that suits the users need's.

2.6 Prototyping

Prototyping is the third activity of the interaction design process where a scaled down manifestation of the final product called a prototype is created [6]. These prototype often comes in varying forms of fidelity with each having their own benefits. Building prototypes has multiple purposes, it encourages the designers explore different

design concepts, it can serve as a fallback when discussing ideas with stakeholders and it can provide a base for usability testing [6].

2.6.1 Low Fidelity Prototype

Low-fidelity (Lo-Fi) prototyping is the concept of making prototypes with limited fidelity. The prototyping is usually done as a analog process with materials such as paper to make the process as fast and cost efficient as possible [22]. Because of the cost efficiency and speed of Lo-Fi prototyping its advantages becomes being able to quickly iterate the design and the possibility to have access to it early in the design process [9]. Making a Lo-Fi prototype is a quite simple process but it still needs some refection and planning. It is important to have all the materials gathered before starting so that the process is not limited by the availability of the materials. It is also important to not only draw illustrations and instead make separable controllable components [22].

2.6.2 High Fidelity Prototype

As the name suggest high fidelity (High-Fi) prototypes have a higher fidelity. There are more alike the finished product and contains more interactions and functionality than the low-fidelity counterpart [23]. The advantages of a high-fi prototype is that the user have more potential to explore the prototype, it is easier to test and often could serve as basis to start writing code from [23]. The advantages from a complete high-fi prototype is great but the drawback is the effort and cost needed to develop one. It is also hard to develop multiple concepts due to these drawbacks which means that there is less exploration of designs [23].

2.6.3 Medium Fidelity Prototype

A Medium fidelity (Mid-Fi) prototype is a level of prototype situated between the previously defined prototypes Low-Fi and High-Fi [24]. Mid-Fi prototypes includes detailed information of navigation, layout, content and functionality but still only uses simple visual design [24]. It still has the advantages of having a lower cost to create than the High-Fi but it still lacks the ability to visualise the final product as with Low-Fi prototypes [24].

2.7 Requirements

According to Lauesen [25] specifying requirements is one of the most important and difficult activity of system development. A requirement is a documented description or need of how the system should perform. Requirements are often produced early in the development cycle and a set of requirements is often used when both designing and implementing a product or system. Requirements can be used for various purposes such as, verification, validation or tracing. Verification is when a check is made to ensure that the product follows the specified requirements. Validation is made to ensure that the requirements correlates to the customer or users needs and tracing is meant to analyse the process either forward or backwards from requirements to implementation.

2.8 Technical Background

This project can on the technical side be split into three different parts. A front-end application which serves the purpose of providing the end users in the company with a presentation of the data gathered in the project. A database server which extracts the data from all the customers databases and stores it in one place called a warehouse. A application programming interface (API) which acts as a middleware between the database server and the front-end application.

2.8.1 React

React is a popular front-end JavaScript library for writing and building modern user interfaces developed by Meta [26]. One of the many benefits by using React is that it is both free and open source, which makes it very easy to pickup. React uses an approach called component based software engineering which lets the developer split the interface into reusable and independent entities [27]. This means that similar interface entities often uses the same code with minor adjustable variations. React offers their own syntax called JavaScript Syntax Extension (JSX) which enables the developer to write both JavaScript and markup in the same file, removing the necessity of separating the markup from the logic [27]. The syntax therefore supports both JavaScript and HTML tags.

2.8.2 .NET & ASP.NET

.NET is a software development platform made by Microsoft which support various programming languages, libraries and other tools [28]. ASP.NET is an open source

framework of the .NET platform that enables the creation of web applications. The ASP.NET framework supports the creating of many different types of web applications ranging from web pages to application programming interfaces (API) [28]. The ASP.NET framework support multiple operating systems and can be developed and deployed on many different platforms such as Windows, Linux and macOS [29].

2.8.3 Microsoft SQL Server

Microsoft SQL Server is a Relational Database Management System (RDMS) that is part of Microsoft's Data Platform and provides support for working with relational databases [30]. As the name states the SQL Server uses an proprietary extension of the Structured Query Language (SQL) which is a common language for querying relational databases [31].

3 Investigation Phase

This chapter describes the investigation phase which is the first phase of the process. The investigation phase primarily relates to the first activity in the interaction design process, establish requirements. This by collecting data for establishing the users, create a conceptual model and then defining the initial requirements.

Figure 3.1: The current phase of the process, investigation.

3.1 Interviews

For gathering data in the project we primarily performed interviews with stakeholders in the company. Because the product developed was only going to be used internally in the company no external persons were interviewed. For the interviews an unstructured approach were taken. This because as stated in the theoretical chapter, open ended interviews are often used in early in the life cycle to explore a greater amount of topics and to find out more about the users motivations and backgrounds. The interviews was divided into three parts were the first part contained questions regarding the background and demographic of the user. Here we wanted primarily to get information regarding the experience of the interviewee. The second part of the interview consisted of questions regarding what the interviewee wanted from the application to be able to rely on it for data driven decisions. This consisted of questions on what data and information the interviewee needed from the customers. The third part of the interview tackled how the interviewee wanted the data to be displayed and their previous experience with an older and more primitive tool used currently in the company. In total six employees from Grade were interviewed and

the interviewees were five males aged 27, 33, 38, 39, and 50 and one female with the age of 38, which in total lead to an average age of 37. Interviewees were also asked about their technical experience on a scale of one to five were the average for all of the interviewees was 3.8 on the scale.

3.2 Affinity Diagram

For processing the large amount of qualitative data gathered from the unstructured interviews an affinity diagram was used. Because affinity diagrams is built bottom up the first result was a uncategorized diagram as seen in figure 3.2. The diagram was then sorted into the following categories: views, logic & automation, data, interaction and issues. These categories were not predetermined and naturally appeared when sorting through the data. The process was done alone due to no other designers being available and the finished diagram can be seen in figure 3.3

Figure 3.2: Uncategorized items

The view category contains all proposed views or pages for the end-product from the interviews which includes views such as homepage or a list of customers. The logic and automation category contains proposed ideas of what the application should run in the background or provide automatically to the user. The data category gathers all suggestions on what data they want available in the end-product. The Interaction category is the interviewees desires on the end-product interactions. The last category issues is the problems the interviewee had with the old product.

3.3 Personas

From the data gathered in the interviews personas was also created. In total four personas were created, one for each departments interviewed in the company. This be-

Figure 3.3: Completed Affinity Diagram

cause of the wide range of use scenarios and demographics each of these departments inherits. The personas included a fictional information surrounding background, motivation, demographics and experience based on data from the interviews. Three of the personas created were males because of the demographics of the specified departments and based on the demographics from the interviews. For the pictures in the personas AI generated images was used from the tool *Generated Photos* [32]. The created personas is shown in figures 3.4, 3.5, 3.6 and 3.7.

Erik

Software Developer

BACKGROUND

With a considerable interest in technology Erik cherishes his role as a software developer. Erik is a junior software developer and has only worked for his current employer after graduating with a degree in computer science. In his spare time Erik spends time with his friends either by participating in sports or playing computer games.

ABOUT

GENDER: MALE
AGE: 25
ROLE: Software Developer
EDUCATION: Degree Computer Science
LOCATION: Lund, Sweden

MOTIVATION

In Eriks role as a software developer he often makes changes in the product which could impact customers in different ways. Erik finds it difficult to predict which and how many customers will be impacted by the changes he makes. Therefore Erik would benefit from a tool that presents this information to him.

EXPERIENCE

TECH ●●●●●
DATA ●●●●○

Figure 3.4: Description of a fictional person in the developer team

Anna

Customer Success Manager

BACKGROUND

Anna's passion for teaching comes in handy for her current role in customer success where she helps customers succeed with the software product produced by her company. When Anna is not working she prefers to read books and spend time outside with her spouse and two children.

ABOUT

GENDER: FEMALE
AGE: 36
ROLE: Customer Success Manager
EDUCATION: Degree in Teaching
LOCATION: Lund, Sweden

MOTIVATION

In Anna's role in Customer Success she often needs to present data to the customers which is currently done in an annoying manual process which steals time from Anna which she could spend on helping other customers. Anna would like to have a tool to view and extract data which she can use for her own decisions

EXPERIENCE

TECH ●●●○○
DATA ●●○○○

Figure 3.5: Description of a fictional person in customer success

Peter

Sales/Marketing

BACKGROUND

With his many years of experience in sales and marketing Peter finds comfort in his current role. Peter might now have the highest experience with IT but that he makes up with his eagerness to learn. When Peter is not working he prefers to go on trips with his spouse and two children, listen to music or play a round of golf.

MOTIVATION

To better succeed in his current role Peter wants to know what and how much the customers are using the product and to be able to compare it with what they have included in their contract. This to easily propose new solutions for his customers which suits the customer better and hopefully bring more revenue to the company.

ABOUT

GENDER: MALE
AGE: 45
ROLE: Sales/Marketing
EDUCATION: Degree in Business & Economics
LOCATION: Lund, Sweden

EXPERIENCE

TECH ●●○○○
DATA ●●●○○

Figure 3.6: Description of a fictional person in sales/marketing

Figure 3.7: Description of a fictional person in the support

3.4 Scenarios

With the created personas scenarios were written, each scenario was written in a three step structure with each of the personas as an actor. Therefore a total of four scenarios were created.

Scenario 1

Step 1: Erik finds himself wanting to change a module in the software but does not know which customers are affected.

Step 2: Eric uses the application to find out which customers are using the module he intends to change.

Step 3: Eric finds the answer he was looking for and can now keep on developing.

Scenario 2

Step 1: Anna wants to see if the customer she is helping is using a specific module.

Step 2: Anna uses the application to find out see which modules the customer is

using.

Step 3: With the information Anna now has she can provide better help to her customer.

Scenario 3

Step 1: A customer uses modules in the product that were not allocated in their contract.

Step 2: Peter sees that the customer is using the module in the application.

Step 3: Peter now has the information that the customer might be interested in including that module in their new contract.

Scenario 4

Step 1: Mikael gets a request from a customer regarding a potential bug in the product and wants to know which other customers could also be affected by the bug.

Step 2: Mikael resorts to the application to find the specified module and then which customers are using it.

Step 3: With the knowledge of which other customers could be impacted by the bug Mikael how important and widespread the issue could be

3.5 Requirements

Before entering the prototype phase a set of initial requirements had to be specified. This to help both designing the prototypes and also when implementing the software for the web-application. The requirements were created with the input gathered from the interviews along with continuous discussions with the stakeholders. The first iteration of the agreed upon requirements is presented below:

Customers

REQ 1-1: The solution should contain a view where the user can see all of the customers. The customers should also be searchable and sortable.

REQ 1-2: The solution should contain a view for looking at a customer specified by the customer.

Contracts

REQ 2-1: The solution should contain a view where the user can see all of the contracts. The contracts should also be searchable and sortable.

REQ 2-2: The solution should contain a view for looking at a contract specified by the customer.

REQ 2-3: The solution should offer the user the ability to create or edit contracts.

Features

REQ 3-1: The solution should contain a view where the user can see all of the features. The features should also be searchable and sortable.

REQ 3-2: The solution should contain a view for looking at a feature specified by the customer.

Navigation

REQ 4-1: The solution should have easy to access navigation for the views of the higher order, home, customers, contracts and features.

REQ 4-2: The navigation system should offer a way of correcting navigation mistakes that the user makes.

Home

REQ 4-1: The solution should have a landing page for the user to start their browsing from. ´

REQ 4-2: The landing page should have a shortcut system for the views, customer, contract and features.

REQ 4-3: The user should be able to make changes to the shortcut system by selecting and deselecting items in the system.

3.6 PBS

To provide a hierarchical overview of the solution and to make it easier to see what needs to be developed in the later implementation phase, a PBS diagram was created.The PBS highlights the views specified in the requirements along with the related functions and data they should offer. The green items in the PBS is items or views, the red is functionality and the yellow items is the related data. As specified in the PBS in figure 3.8, the solution contains the views specified by the requirements. The Home Page is the landing page when browsing the solution, and through the navigation the user can visit the other views. This PBS only specifies the items related to the front-end application and does not include the database and back-end solution.

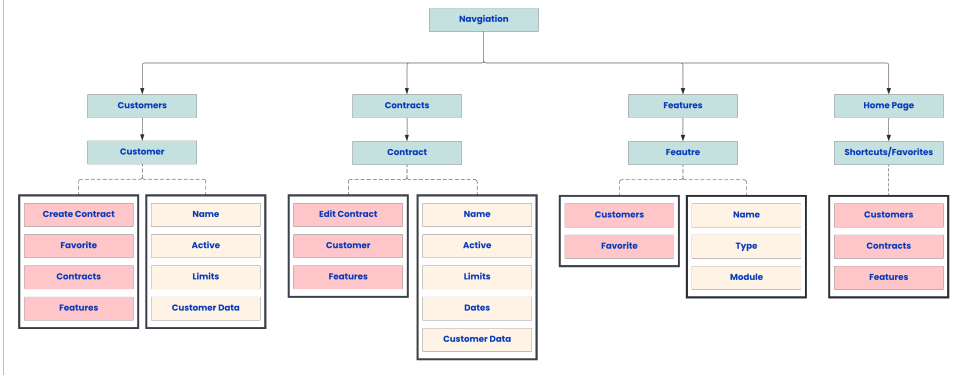


Figure 3.8: The Project Breakdown Structure (PBS) of the proposed solution.

4 Prototyping Phase

This chapter presents the second phase in the process, the prototyping phase. With the established requirements and the conceptual model from the investigation phase, prototyping started. The prototyping was split into three parts, a Low-Fi, Mid-Fi and High-Fi prototype which was created sequentially with each type of prototype going through its own iterations.

Figure 4.1: The current phase of the process, prototyping.

4.1 Low-Fi Prototype

The prototyping phase started by creating a Low-Fi prototype with regards to the conceptual model and requirements specified in the last phase. The Low-Fi prototype were meant as a tool to minimize discrepancies between the designers and stakeholders, and to test basic interactions and functionality such as navigation and flow of the product. Because Low-Fi prototypes need to be fast and inexpensive the prototype was created using basic office supplies, such as pen, paper and tape. When creating the design for the Low-Fi prototype some of the previously described design principles was kept in mind such as using the affordance of buttons for clickable entities. A total of six primary views were created with several smaller modals, which are presented in figure 4.2 to 4.8.

4.1.1 Evaluation

In total five Low-Fi tests were conducted with at least one test user from every internal department. The five participants consisted of four males and one female and they had an average age of 38 years. Each user had prior experience with the Grade platform and some had experience with a prior solution. The observations were conducted in a controlled environment and users were also instructed to follow the think-aloud technique, both described in the theoretical chapter. During the observations the users were instructed to follow a set of scenarios presented in table 4.1 and most of the observations took around 15 minutes to complete. After the observation the test users were also asked for general opinions on the prototype. All thoughts from the think-aloud technique and questions and actions were documented.

Table 4.1: The scenarios used in the Lo-Fi evaluation.

Scenario	**Summary**
Scenario 1	Lookup if Customer - 1 is close to the current contract limit
Scenario 2	Lookup which features that Customer - 3 has enabled
Scenario 3	Lookup which customers that has Feature - 1 enabled
Scenario 4	Lookup the currently active contract for Customer - 1
Scenario 5	Create a new contract for Customer - 1
Scenario 6	Check if Customer - 2 has any expiring contracts
Scenario 7	Check if the are any expiring contracts

4.1.2 Feedback

The Low-Fi evaluations did not reveal any glaring problems with the prototype but it did reveal some issues the user were having. One of the most apparent issue was with the navigation flow of the application, some of the test users had issues with finding the right pre-planned path to the specified view or modal. This resulted in that the users lost themselves in the prototype and had to ask for help from the observer. Some test users suggested that the problem did not arise from the the navigational flow of the prototype but due to the primitiveness and the time it took to switch views. The majority of the users also mentioned that the term *module* which is used in the module view were users can view information on what menu and settings the customer is using was a bit confusing. This because of how Grade use the term internally in the company. Suggestions of an alternative name was provided and the most prominent was the terms *functions* or *features*. Users also reported that the

favorite shortcut for contracts, see figure 4.2, was unnecessary and suggested that it would better to replace it with a list of the contracts that are closest to expiring. Some test users also thought that the extra filter for customers in figure 4.3 was also unnecessary and having the search-bar combined with the sortable columns was more than enough for finding the correct customers in the list. The evaluation also provided the feedback that the prototype lacked functionality for creating new customers and some users suggested that it could be done by adding button for this in the listing of customers in figure 4.3 along with a modal. Below is a short summary for the feedback presented:

Navigation

- Users getting lost and not knowing where they are in the prototype.
- Users not finding a path to complete the presented task.

Module View

- Users finding the naming of the view confusing because of existing terminology in the company.
- Suggestions to change it to function or feature.

Shortcuts

- Change the logic of the contract shortcut to soon to expire instead of favorites.

List Views

- Remove extra filtering because the sortable columns and search bar is enough.
- Add functionality for creating customers

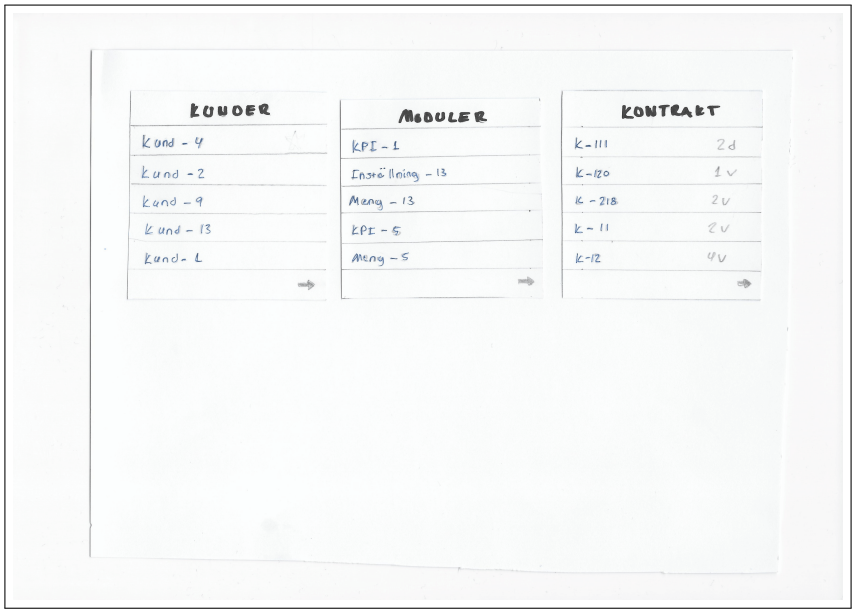


Figure 4.2: The landing page of the Low-Fi prototype, from the image the shortcuts to customers, contracts and features can be observed.

Figure 4.3: A list view for the Low-Fi used for the customers in the application, which displays the means to search and sort the list.

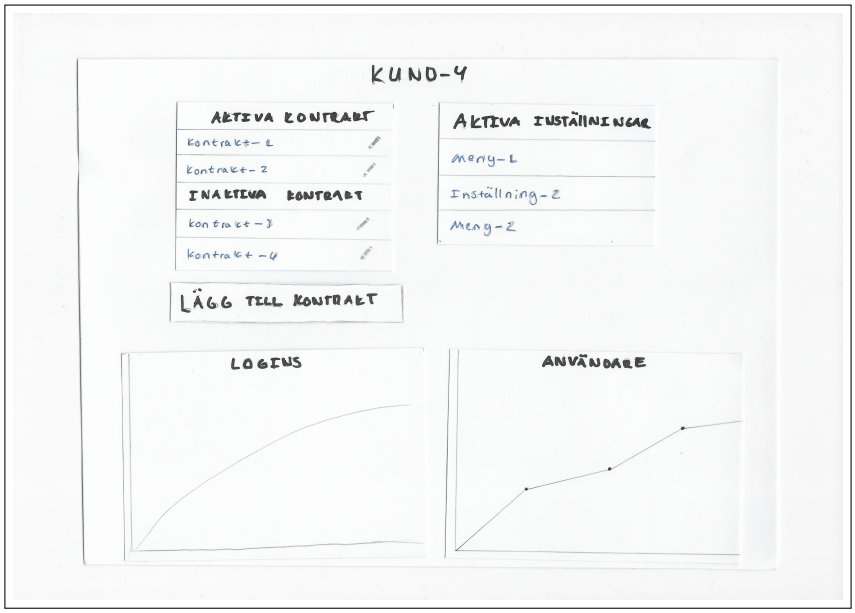


Figure 4.4: A page for viewing a specified customer, which contains data such as active contract etc.

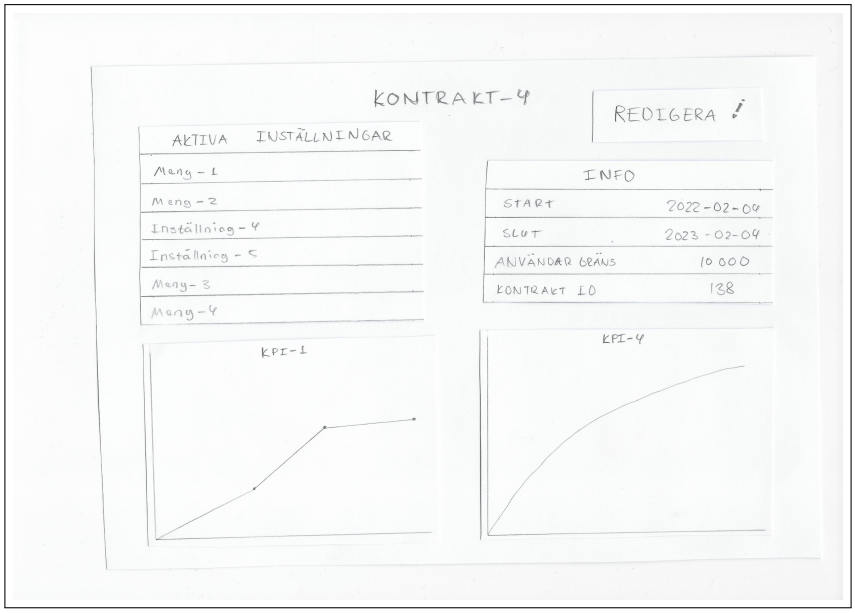


Figure 4.5: A page for viewing a specified contract, which contains data such as active features etc.

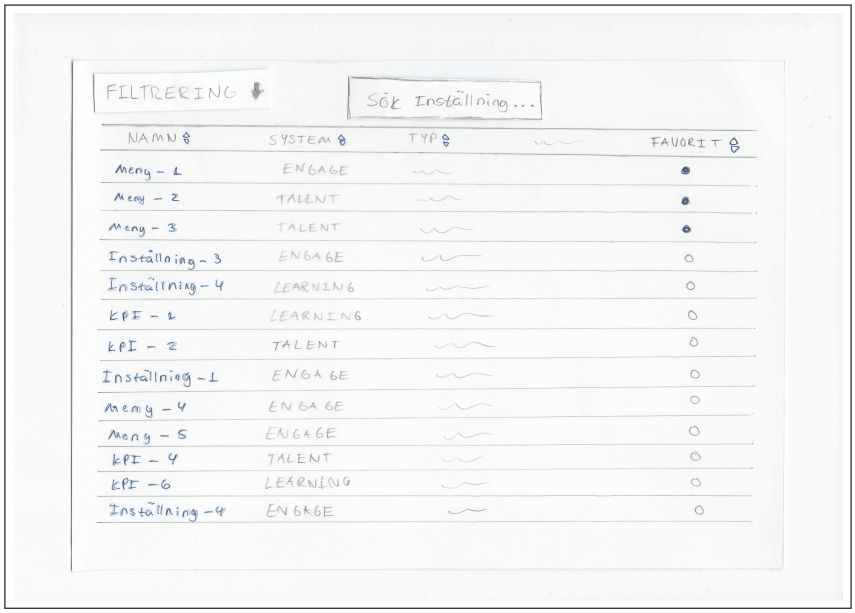


Figure 4.6: A list view in the Low-Fi used for the features in the application, which displays the means to search and sort the list.

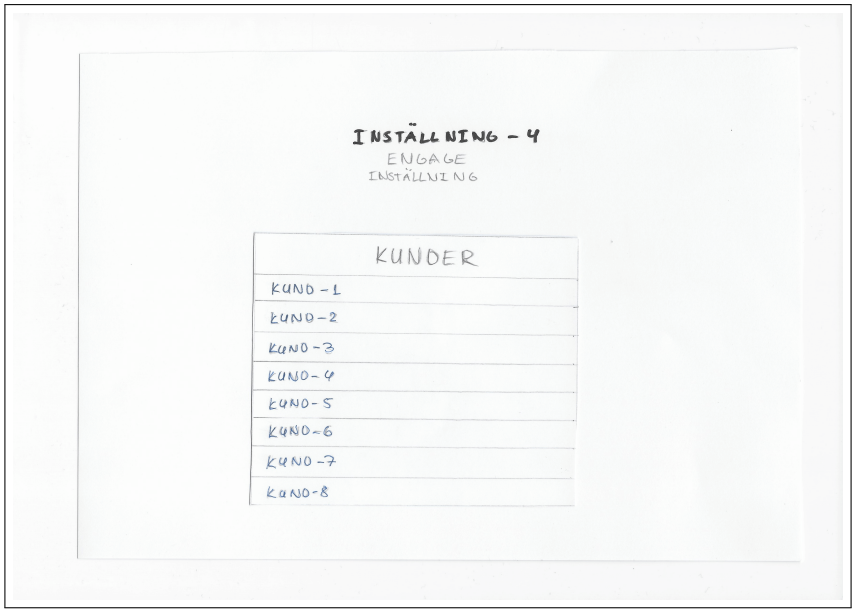


Figure 4.7: A page for viewing a specified feature, here a list of customers using that feautre is displayed.

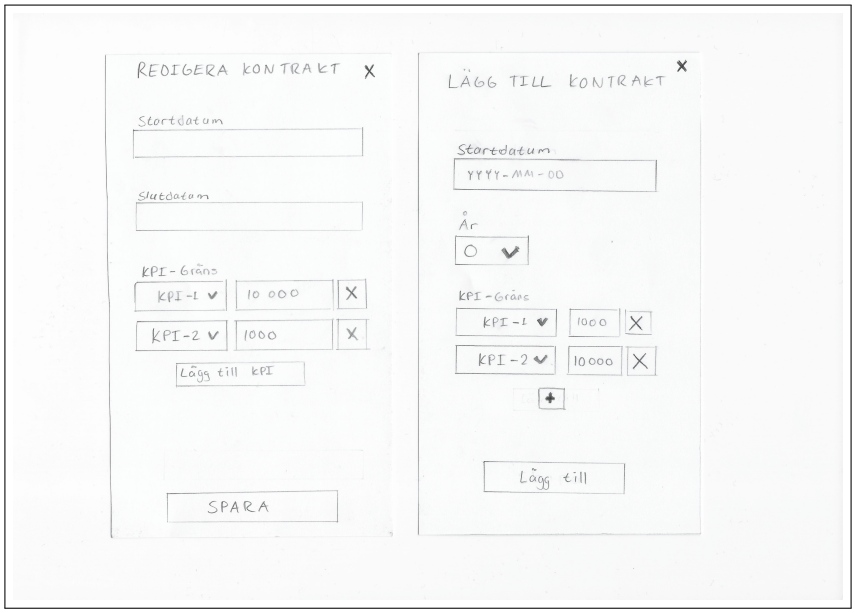


Figure 4.8: Add and edit contract modals for the Low-Fi

4.2 Mid-Fi Prototype

The second part of the prototyping phase is defined by the creation of a Mid-Fi prototype. The main purpose of the Mid-Fi prototype was to improve the flow of the application which was one of the most major issues appearing in the Low-Fi prototype evaluation. The prototype was developed mainly with Figma which allowed to quickly alter the flow between the different views created in the Low-Fi prototype. The final iteration of the Mid-Fi prototype retained the six views and all the minor modals, which can be seen in figures 4.10 to 4.17.

4.2.1 User Flows

To improve the flow when creating the Mid-Fi prototype user flows where created. These flows would serve as a fallback that the designer could utilize when creating the actual Mid-Fi prototype. The user flow would also serve as a form of quick and dirty evaluation of the flows that existed in the Low-Fi prototype to find potential issues. A couple of flows where setup from the scenarios the test users performed in the Low-Fi prototype evaluation, the flow of adding a contract for an existing customer can be seen in figure 4.9

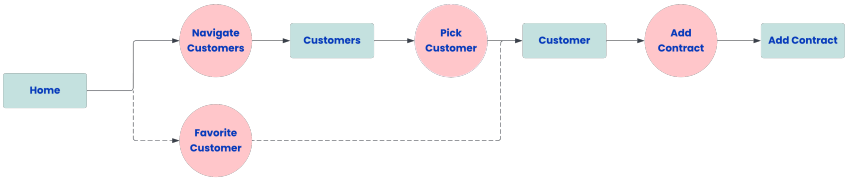


Figure 4.9: User flow diagram for adding a contract to an existing customer

4.2.2 Evaluation

The Mid-Fi prototype was evaluated using observations with the think-aloud technique which overall gave a positive result. The prototype was evaluated on four people in different departments. The participants consisted of three males and one female with a combined average age of 35 years. Similar to the Low-Fi prototype every test person had prior experience with the GRADE platform. The observation was again in a controlled environment and was tested on the same computer in all tests. During the observation the test users got nine scenarios (Table 4.2) described to them and was asked to perform them. The users was also asked to think loudly as instructed by the think-aloud technique and their thoughts were taken down as notes. After the scenarios the users also got time to freely explore the product while still expressing their thoughts, which also were documented as notes. Most of the observations took around 20 minutes with the scenarios consuming most of the time.

Table 4.2: The scenarios used in the Mid-Fi evaluation.

Scenario	**Summary**
Scenario 1	Lookup if Customer - 1 is close to a current contract limit
Scenario 2	Lookup which features that Customer - 3 has enabled
Scenario 3	Lookup which customers that has Feature - 1 enabled
Scenario 4	Lookup the currently active contract for Customer - 1
Scenario 5	Create a new contract for Customer - 1
Scenario 6	Check if Customer - 2 has any expiring contracts
Scenario 7	Check if the are any expiring contracts
Scenario 8	Add Customer - 3 to favorites

4.2.3 Feedback

The evaluation of the prototype revealed that the navigation problem existing in the Low-Fi prototype was not as prominent in the Mid-Fi variant. Because no major changes in navigation was performed between the last iteration of the Low-Fi prototype and the first iteration of the Mid-Fi prototype this could have been accounted to other factors. The observed users had less problems with finding the correct path and did not get stuck in the same way as with the Low-Fi prototype. One issue the Mid-Fi revealed was that the test users felt that the prototype lacked combined data from all customers that could be used as an overview for the whole GRADE platform. The consensus was that this could be placed on the home screen below or above the favorite shortcuts in figure 4.10. Another issue was that the priority of content was lacking for customers and contracts, some test users thought that they had to look through to much of the information to get to the data they wanted, for example in figure 4.14 a test person wanted to have the data of the contract before its meta-information so they could retrieve it quicker. Some users also wanted to have a quicker access to adding contracts and customers, and suggested a button for this directly in the list of customers and contracts. Another issues with the prototype was that there were lacking information about the contract limits a customer has, both in the list view but also in the direct view of the customer (Figure 4.12) or contract (Figure 4.14). A couple of test users also felt that they did not want to have to go in every contract for a customer to edit it and wanted functionality for editing straight from the customer view (Figure 4.12). Test users also reported that it was hard to keep track on which page they currently were on and suggested to add color to the page on the navigation bar, which can be seen at the top of figure 4.10. Below is a short summary of the feedback from the Mid-Fi prototype presented:

Navigation

- Users are not having the same problems with getting lost in the prototype.
- Users are not having the same problems with finding the flow of the navigation.
- Add color to the navigation to indicate which page the user are on.

Home

- Users wanted to have data related to all customers in the form of an overview.

Contracts

- Change the priority of content.
- Add button for adding contacts directly in the list.

- Add button for editing contracts in the customer view.

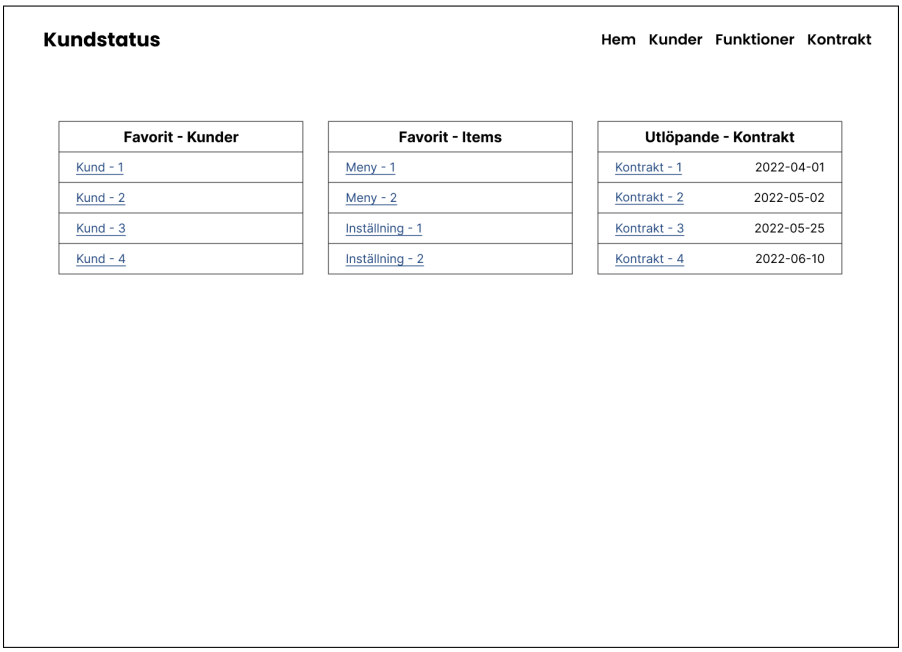


Figure 4.10: Landing page for the Mid-Fi, shortcuts to the other pages can be observed.

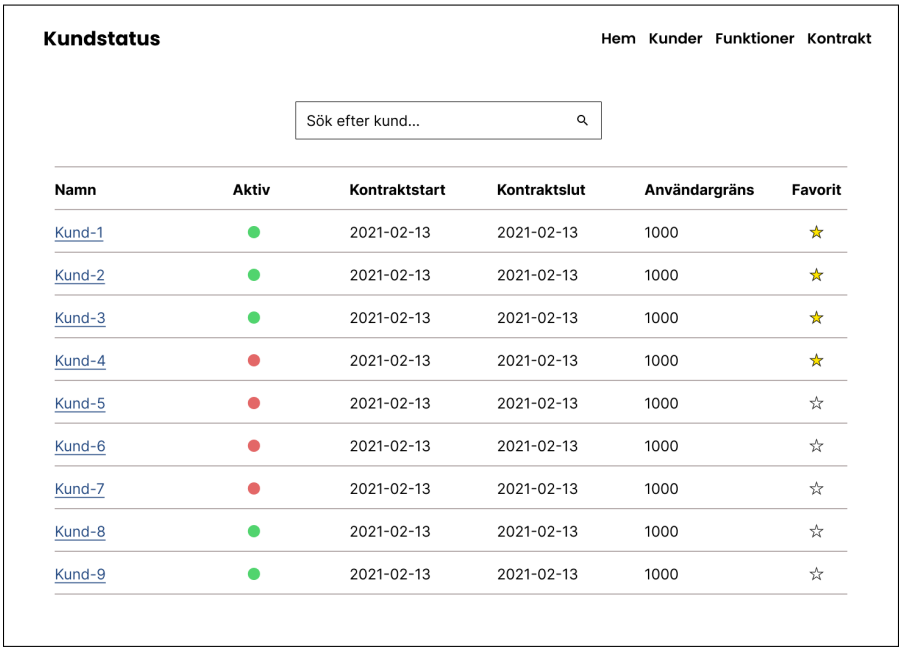


Figure 4.11: List of customers in the Mid-Fi, which displays more information related to the customer.

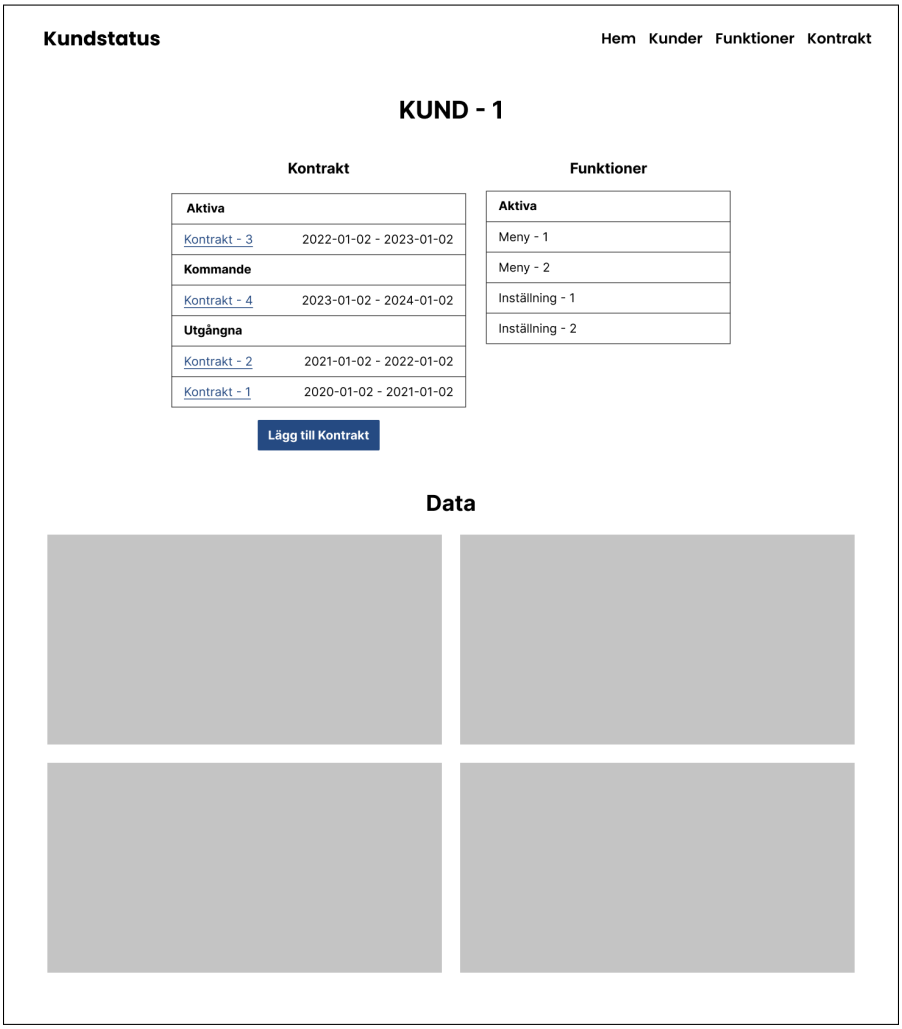
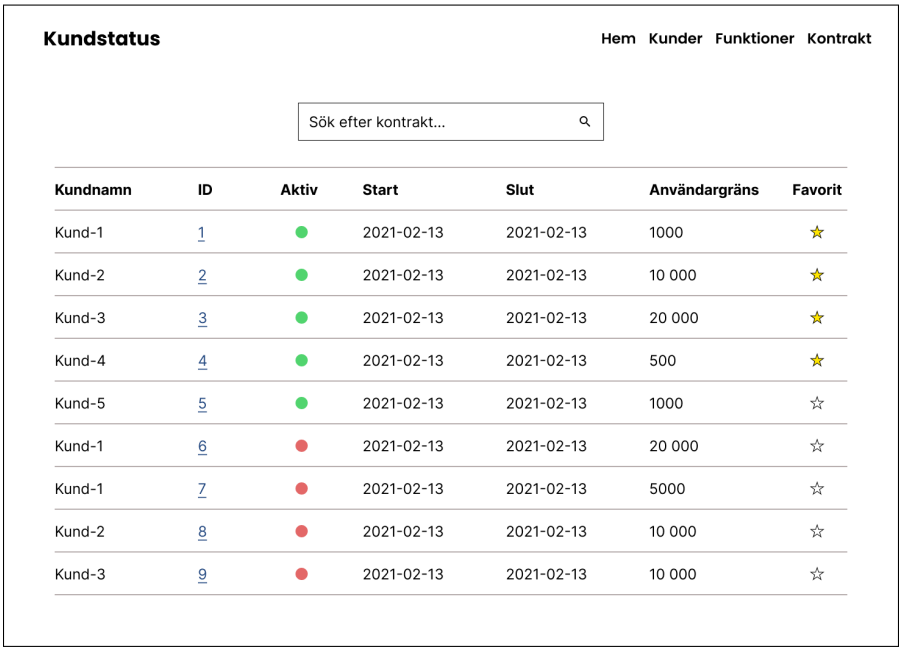


Figure 4.12: A page for viewing the specified customer in the Mid-Fi, its contracts and related customer data.



Kundnamn	ID	Aktiv	Start	Slut	Användargräns	Favorit
Kund-1	1	●	2021-02-13	2021-02-13	1000	★
Kund-2	2	●	2021-02-13	2021-02-13	10 000	★
Kund-3	3	●	2021-02-13	2021-02-13	20 000	★
Kund-4	4	●	2021-02-13	2021-02-13	500	★
Kund-5	5	●	2021-02-13	2021-02-13	1000	☆
Kund-1	6	●	2021-02-13	2021-02-13	20 000	☆
Kund-1	7	●	2021-02-13	2021-02-13	5000	☆
Kund-2	8	●	2021-02-13	2021-02-13	10 000	☆
Kund-3	9	●	2021-02-13	2021-02-13	10 000	☆

Figure 4.13: List of contracts in the Mid-Fi, which displays more information related to the contract.

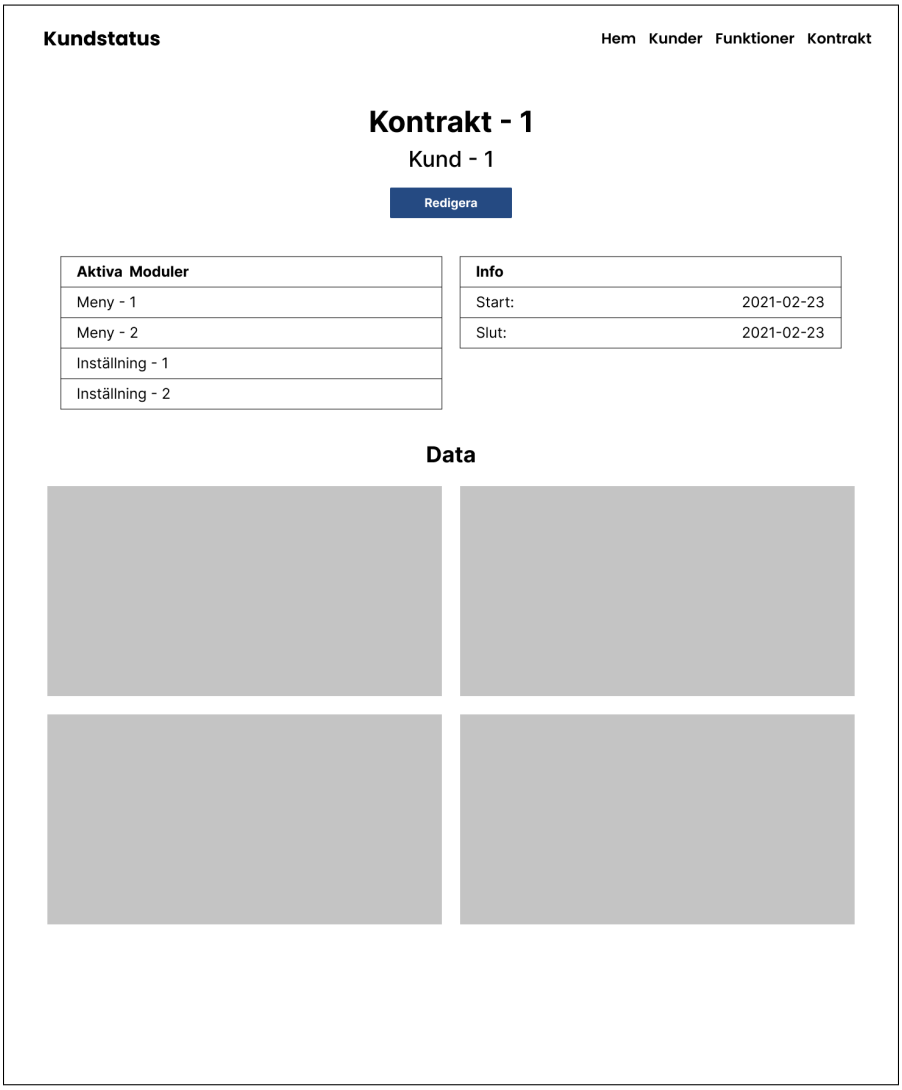


Figure 4.14: A page for viewing the specified contract in the Mid-Fi and its active features.

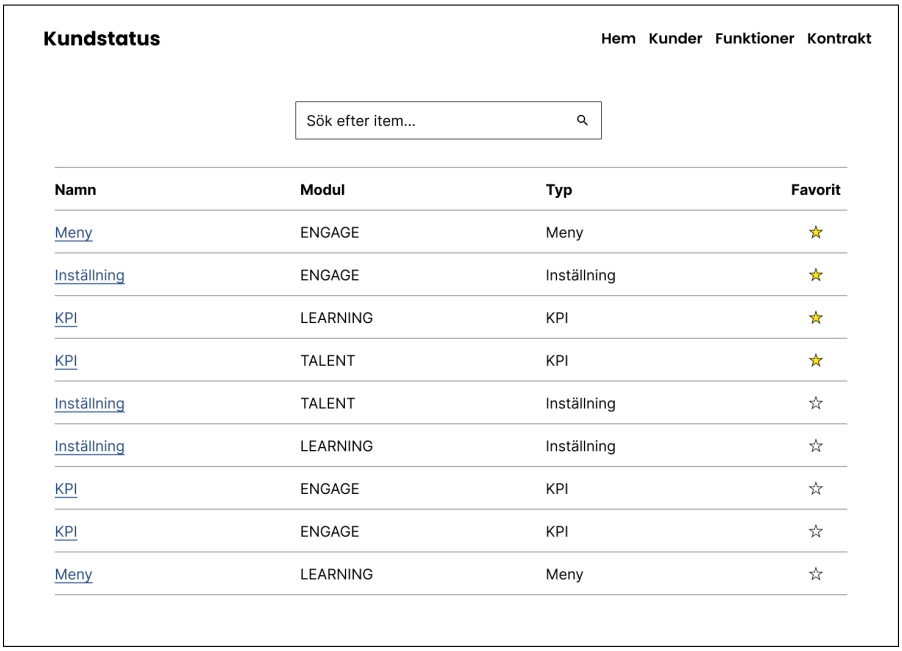


Figure 4.15: List of contracts in the Mid-Fi.

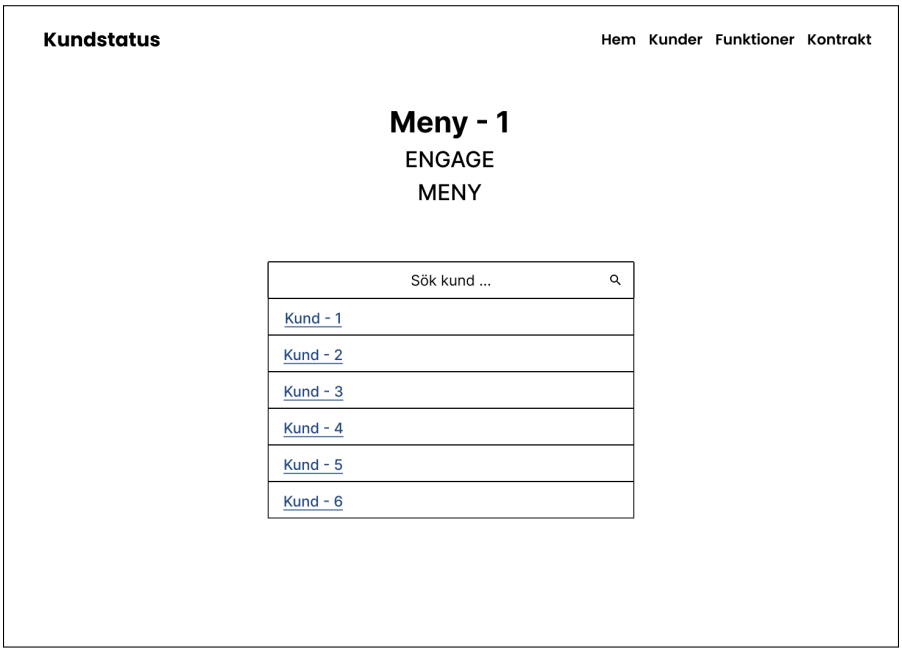


Figure 4.16: A page for viewing the specified feature in the Mid-Fi, shows all the customers related to that feature.

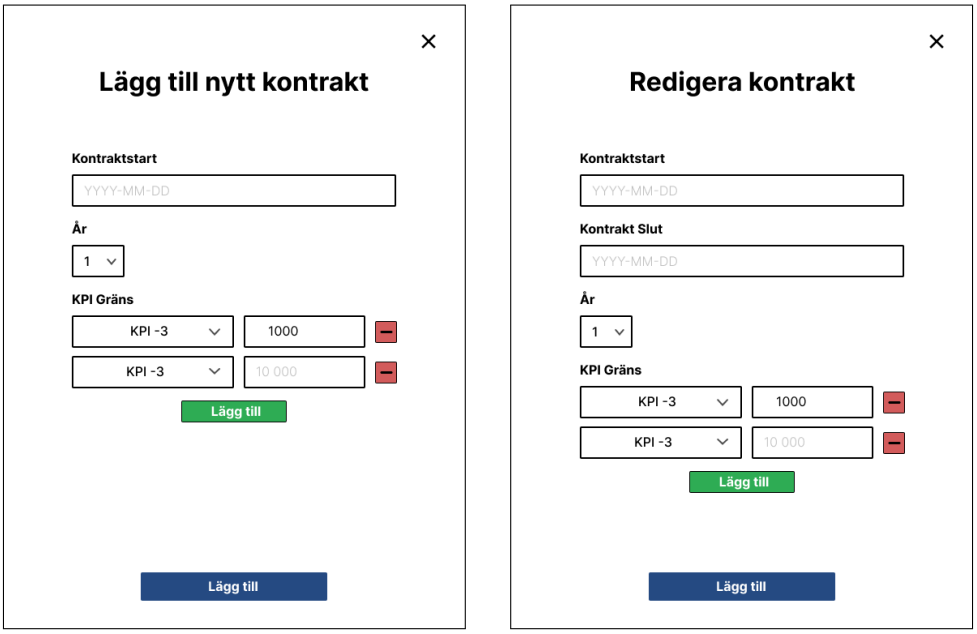


Figure 4.17: Add and Edit contract modals in the Mid-Fi prototype.

4.3 High-Fi Prototype

In the final part of the prototyping phase a Hi-Fi prototype was created. With the feedback received in the previous prototypes a brainstorming session was performed to find solutions to the issues. This session was performed alone, because as stated before it was hard to get hold of other designers to hold it with. The High-Fi prototype was as the Mid-Fi prototype created in the software Figma. The decision to stay with Figma for the High-Fi was mainly because it was possible to reuse and refine parts of the Mid-Fi prototype as a base for the High-Fi prototype. Along with Figma, tools such as Adobe Photoshop and Adobe InDesign was also used for creating certain elements.

4.3.1 Changes

After reviewing the feedback from the evaluation of the Mid-Fi prototype work began on creating a High-Fi prototype. The created High-Fi prototype brought on a couple of major and minor changes to the design of the solution along with some new functionality.

Graphical Guidelines

One of the most apparent changes arriving with the High-Fi prototype was the visual overhaul of the solution. When designing the High-Fi prototype more focus were put on the visual design of the application. Stakeholder thought that the solution would benefit by adapting the company's graphical profile, bringing it closer to the design of other tools and applications in the company, hopefully making the users more comfortable with using the solution. The graphical profile offered by Grade contained many suggestions and guidelines on typefaces, colours, placements etc. A majority of these suggestions were incorporated in the prototype which lead to a bunch of visual changes. Both primary, secondary and background colors were changed along with the typeface of both headings and normal text. The changes made to the visuals of the prototype was also made with the WCA guidelines in mind. All the changes made to the colours and typefaces was made to conform with WCAG.

Alternatives

The High-Fi prototype also came with some alternatives, two versions of the feature view and the customer view was made to test different solutions. The feature view alternatives was with how the view was displayed. The first alternative was to have it as a separate page that the user was redirected to when selecting a feature. The second alternative was presenting a modal directly from the view where the user choose a feature. The difference can be seen in figure 4.18. The two versions of the customer view was to see if it was possible to solve the issue certain users had with the priority of information. The first version build upon the already existing one from the earlier prototypes which is a one page design where all the information about the customer exists on the same page and the user scrolls between it which is shown in figures 4.20 to 4.22. The second version is a tab based approach where the user presses one of the tab buttons to get the information they want displayed which is shown in figure 4.19.

Functionality

Some new functionality was also added with the High-Fi prototype. Test users suggested that there should be a way of adding contracts directly from the contracts view instead of going through customers. To support this a button was added in the contracts view which can be seen in figure 4.25. Some users also suggested the functionality of adding customers directly from the solution which was previously done manually in the database. To fulfill the suggestions a new modal was created for creating and editing customers along with a button in the customers view, these changes can be seen in figure 4.25 and 4.28. Users also felt that they wanted to be able to quicker edit contracts, therefor a edit button directly on the contracts in the customer view as added, see figures 4.20.

Other Changes

The High-Fi prototype also addressed the issues with the lack of content on the home page. An overview was placed above the shortcuts on the home page with data related to all the customers of the company. This can be seen in figure 4.23. As mentioned before some changes was made to fix the prioritisation of the displayed data, for the contract view the main data was placed alongside with the contracts meta-information with the purpose of making the data faster to lookup. This is shown in figure 4.24. A active status was also added in the title so the user did not have to calculate it themselves based on the dates. More data was also added to the list of contracts and customers. All the contract limits was introduced as green or red circles depending on if they were exceeded or not, see figure 4.26.

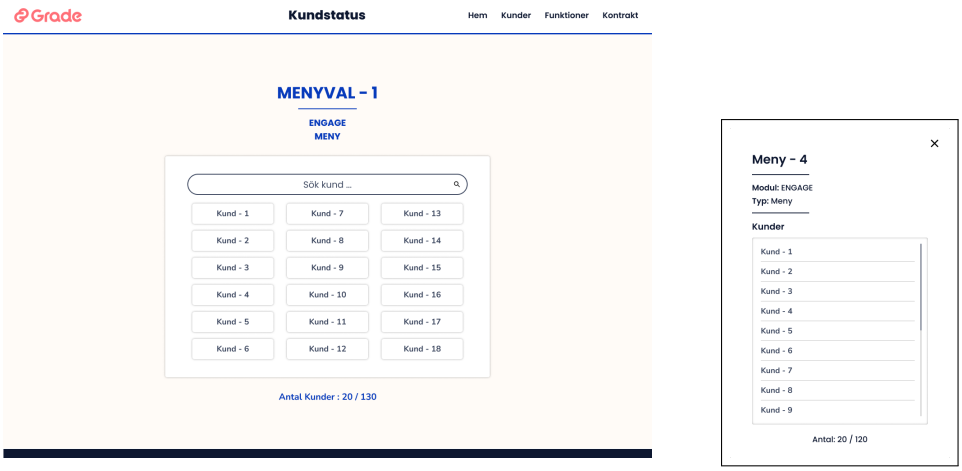


Figure 4.18: Two concepts on how to display the feature view, the left image showing the feature view as a separate page and the right image a modal approach.

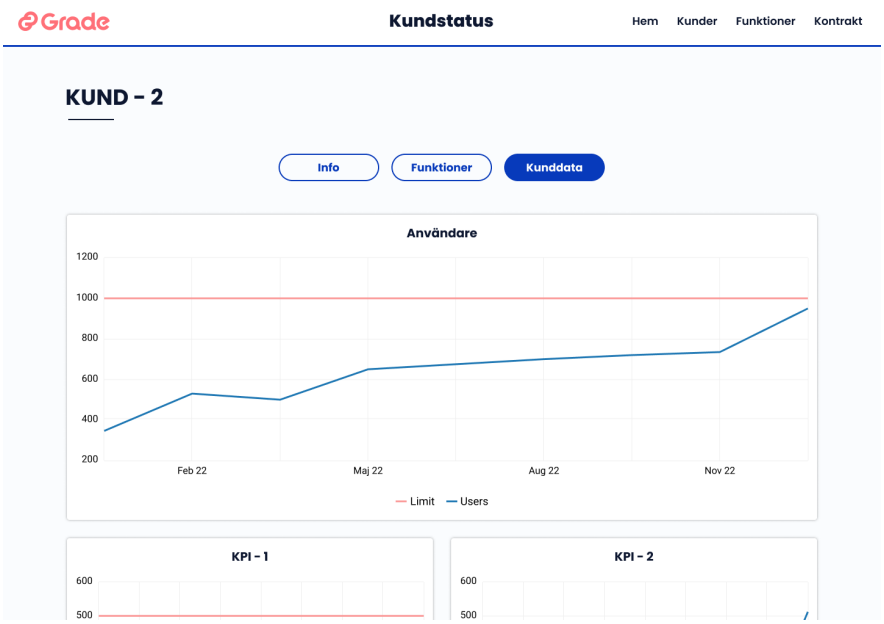


Figure 4.19: The tab based approach for the customer view.

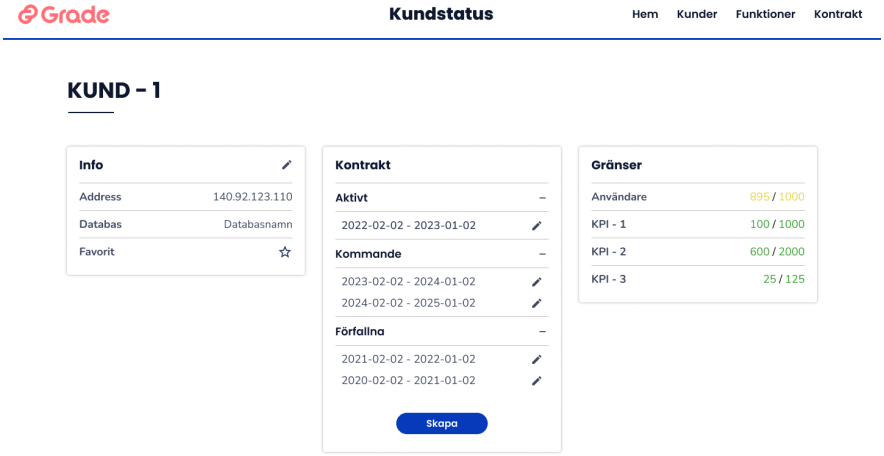


Figure 4.20: The one page approach for the customer view, Part (1/3)

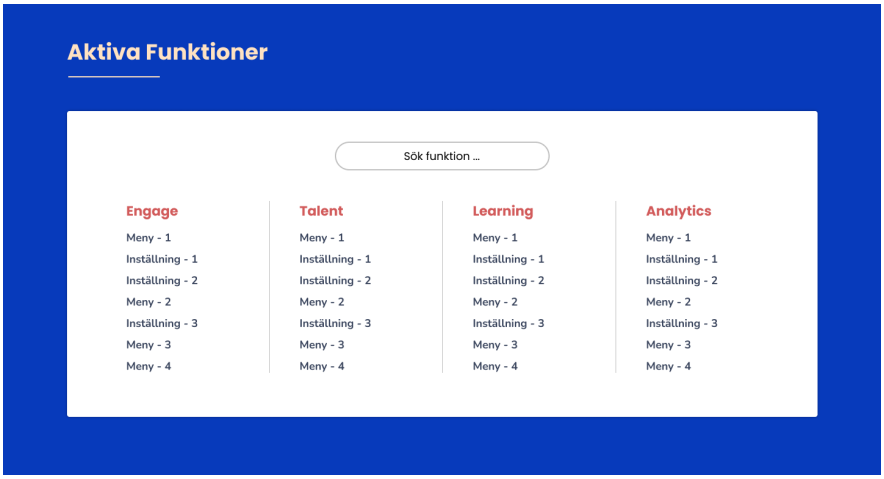


Figure 4.21: The one page approach for the customer view, Part (2/3)

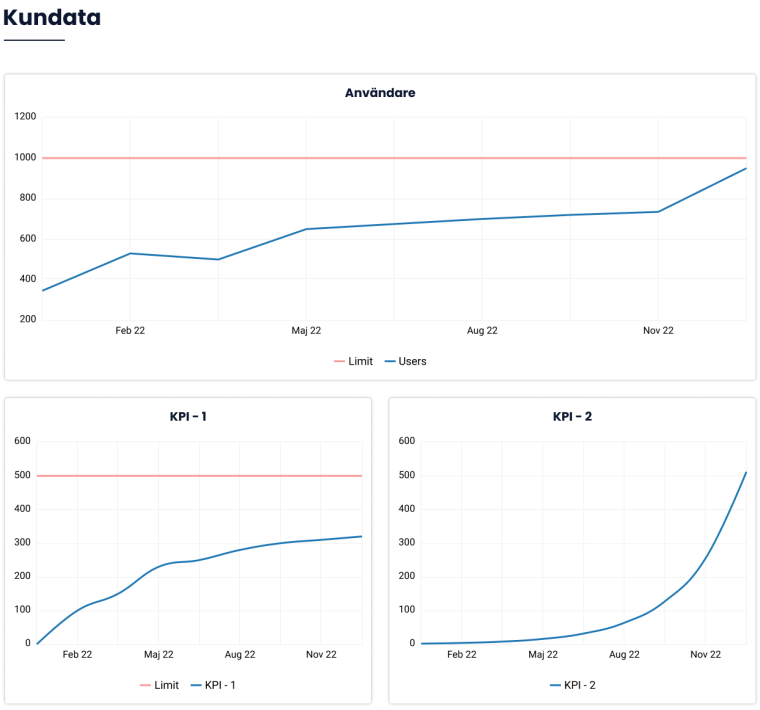


Figure 4.22: The one page approach for the customer view, Part (3/3)

4.3.2 Evaluation

The High-Fi prototype was evaluated using the same process and technique as the Mid-Fi prototype evaluation with the addition of a SUS score and task completion rate. In total four test users participated from different departments in the company along with two external test users. The internal participants consisted of three males and one female with an average combined age of 35 years. The external participants consisted of one male and female with an average age of 36 years. The observations took around 20 minutes and as with the previous evaluation both thoughts, actions and answers to questions were documented. The external users were chosen to take part to get a fresh perspective on the prototype from a user who has not seen the previous iterations, and to see if the application was easy to use with no prior knowledge of Grade and the GRADE platform. Because the external users were not part of the target audience and because they would not use the finished product, SUS questions such as *I think that I would like to use this system frequently* would be hard to answer, therefore the decision was taken to exclude them from answering the SUS questionnaire.

Table 4.3: The scenarios used in the High-Fi evaluation.

Scenario	**Summary**
Scenario 1	Lookup if Customer #1 is close to a current contract limit
Scenario 2	Lookup which features that Customer #1 has enabled
Scenario 3	Lookup which features that Customer #2 has enabled
Scenario 4	Lookup which customers that has Feature #1 enabled
Scenario 5	Lookup which customers that has Feature #2 enabled
Scenario 6	Lookup the currently active contract for Customer #1
Scenario 7	Create a new contract for Customer #1
Scenario 8	Check if Customer #2 has any expiring contracts
Scenario 9	Check if the are any expiring contracts
Scenario 10	Add Customer #3 to favorites shortcut

4.3.3 Result

The result from the High-Fi tests can be split into two categories, qualitative data from the observations and think-aloud method and quantitative data from the scenarios and SUS questionnaire. The result contains data from both the internal and

external users, with the external users only being excluded from the SUS questionnaire.

Qualitative data

For the two customer view concepts there were a unanimous agreement that the one page approach were better. Users thought it was unnecessary to have the extra clicks the tab based approach brought with it for displaying the different data and preferred to scroll instead. In the feature view the majority of the users thought the modal was the better approach, but if the feature viewed was scaled up in the future with more information a full page was more suitable. An issue some users were having was that they did not know where in the table to click to get the information they wanted or that they pressed the wrong column, which could indicate bad use of *signifiers*. Users also felt that they wanted the ability to manually change the end-date for the contracts, which were automated in the last iterations. Some users also felt the lack of a go way of going back when they advanced to a specified customer, contract or feature and suggested a back-button. For the limits in the list views, users felt that they lacked a tool-tip for the circles representing limits and did not want to have go into a customer to see which limit that were exceeded. A user was confused with the colors of the limits and thought that green was when the limit was reached and red when it was not.

Quantitative Data

The result of the SUS questionnaire is shown in table 4.4. The total score for each participant was calculated using formula 4.1, were the variable i controls the questions with positive statements and the variable j controls the questions with negative statements. These scores was then used to calculate an average SUS score, which came to a score of 90.6. The observations also provided a task completion rate which can be seen in figure 4.5. The average completion rate of the provided task came to a percentage of 85%. The first four participants in figure 4.5 is internal and the following two is external. The red task in figure 4.5 for participant five was that the user had problem with finding an active contract and wanted to go through the list instead of through the customer.

$$Score = \left(\sum_{1}^{i} (x(i) - 1) + \sum_{1}^{j} (5 - x(j)) \right) \cdot 2.5 \qquad i = 1,3,5,7,9 \ \& \ j = 2,4,6,8,10 \tag{4.1}$$

Table 4.4: The SUS score for the High-Fi prototype

P/Q	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
Participant 1	4	2	4	1	5	1	5	1	4	1	90
Participant 2	4	1	5	1	4	1	5	1	4	1	92.5
Participant 3	4	1	5	1	5	1	4	1	4	1	92,5
Participant 4	4	2	4	1	4	1	5	1	4	1	87.5

Table 4.5: The task completion rate for the High-Fi prototype.

P/T	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Success
Participant 1											90%
Participant 2											100%
Participant 3											100%
Participant 4											100%
Participant 5											60%
Participant 6											60%

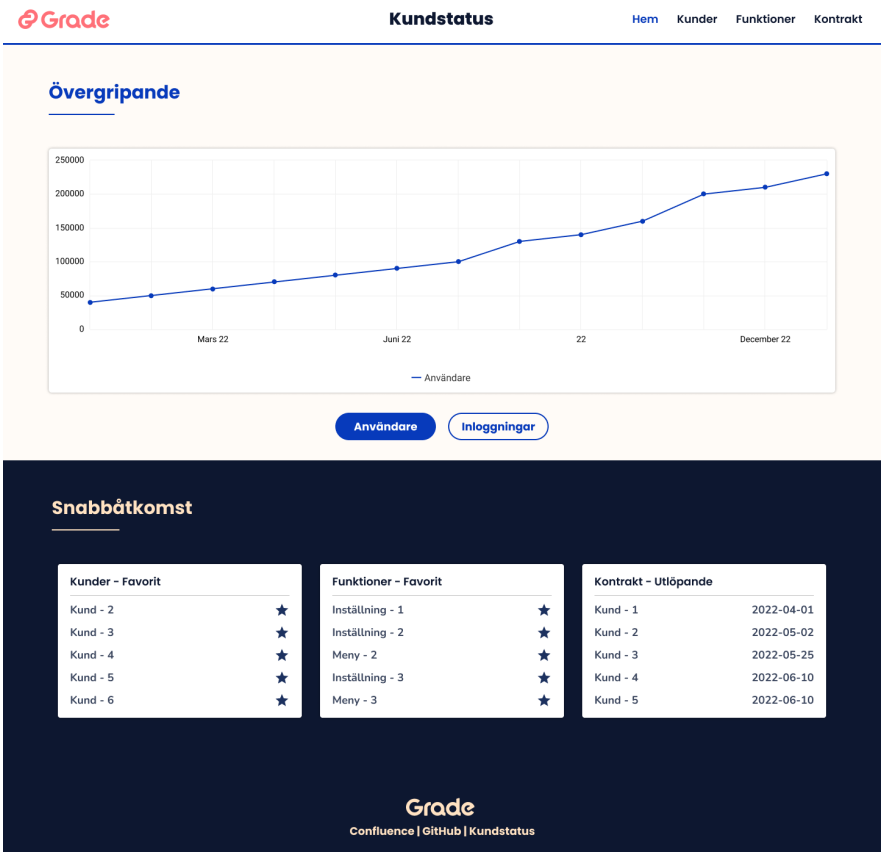


Figure 4.23: The home view in the High-Fi prototype. Here the shortcuts has be updated and the general information has been added.

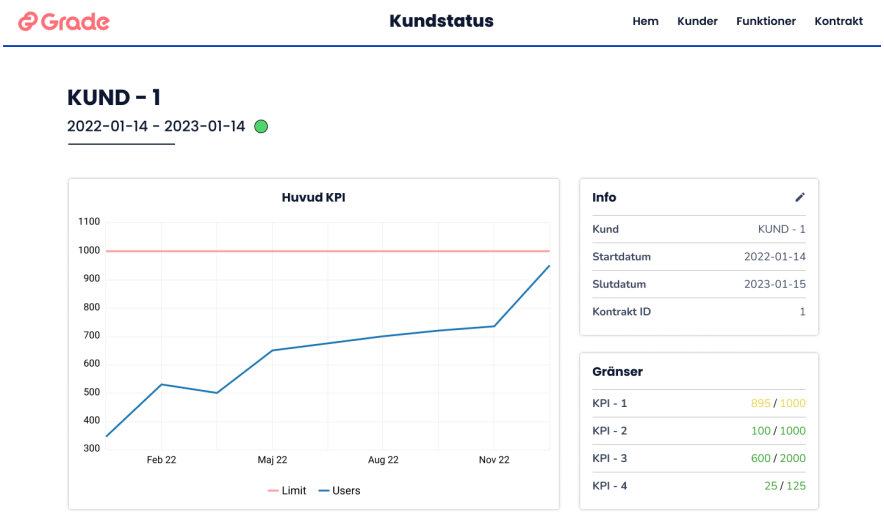


Figure 4.24: The first third of the full contract view, the rest is similar to the customer view.

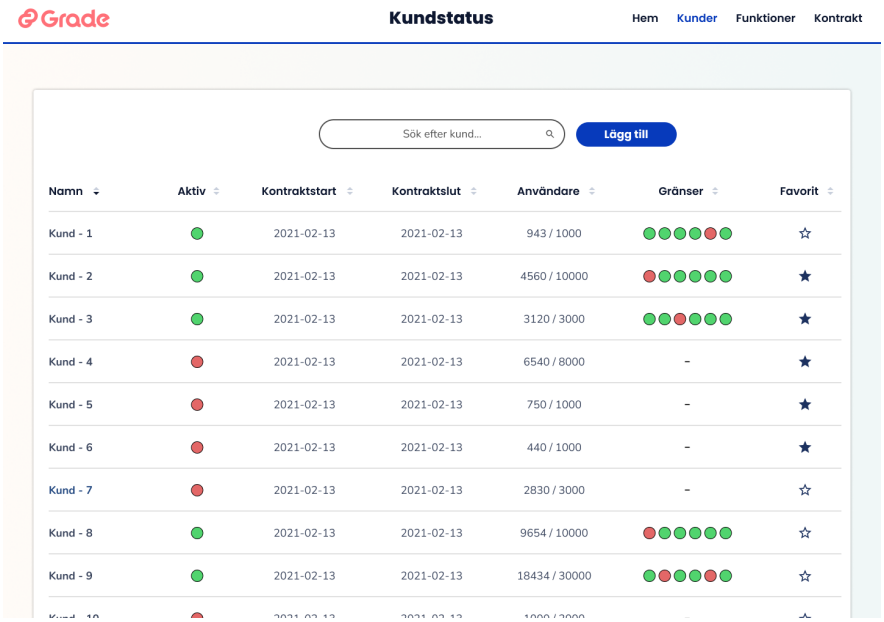


Figure 4.25: The customer list view in the High-Fi prototype.

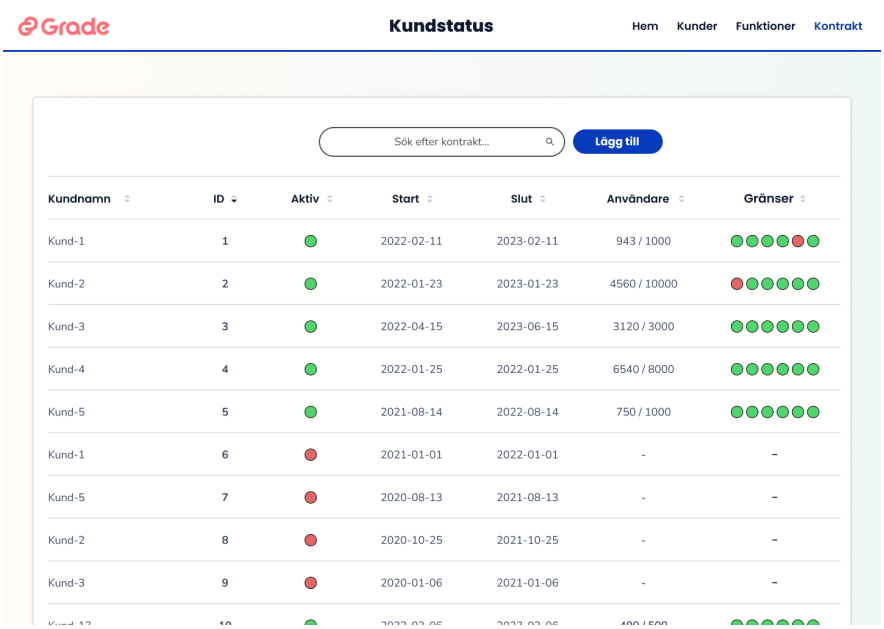


Figure 4.26: The contracts list view in the High-Fi prototype.

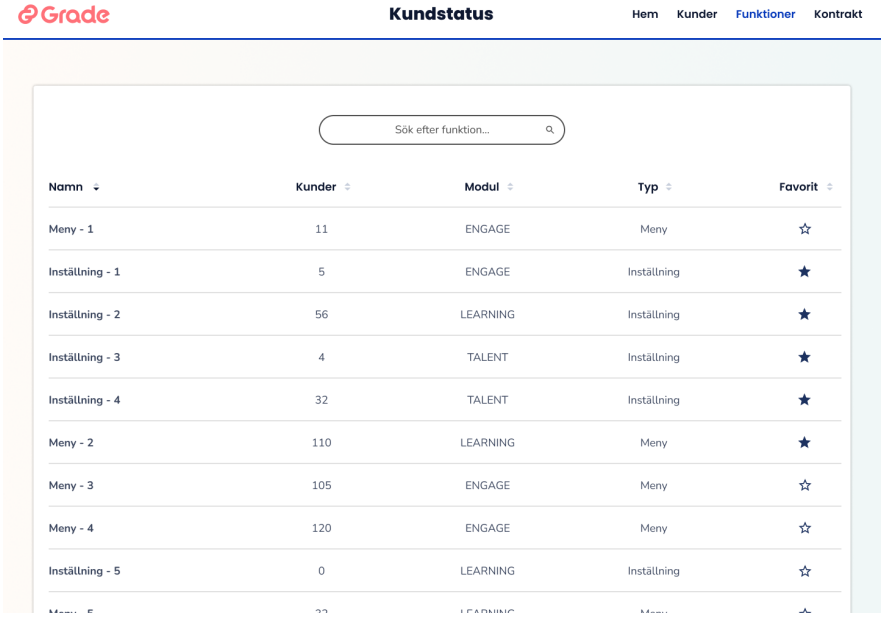


Figure 4.27: The feature list view in the High-Fi prototype

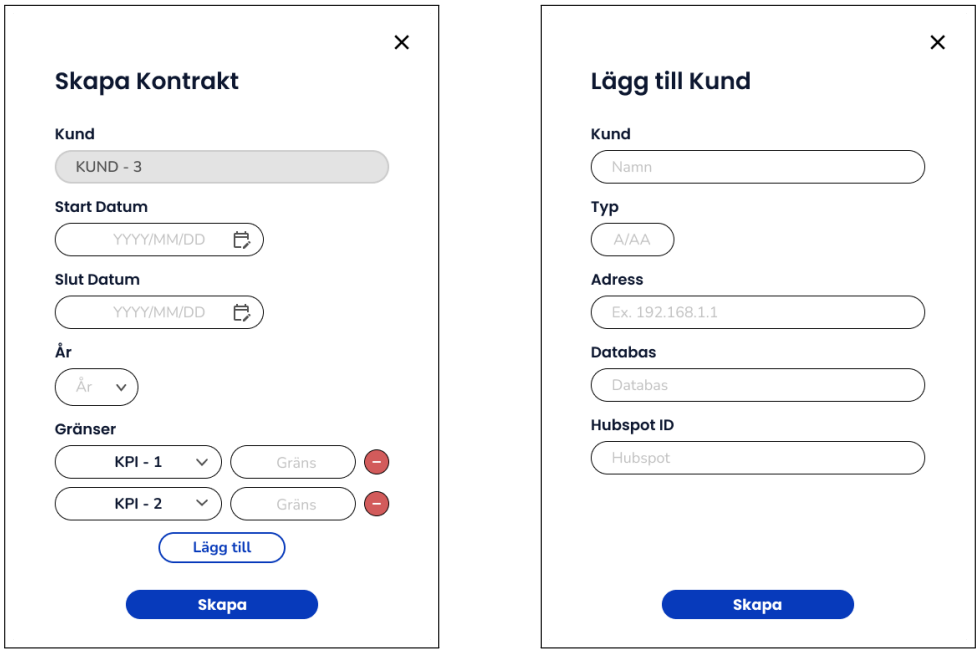


Figure 4.28: The create contract/customer modals for the High-Fi prototype.

5 Implementation Phase

In this chapter the last step of the process, the implementation phase is presented 5.1. This phase includes the steps taken from the final iteration of the prototyping phase to the finished application. The chapter presents technical choices and the results of the implementation along with brief descriptions of the architecture.

Figure 5.1: The current phase of the process, implementation.

5.1 Preparation

The implementation can be broken down into three parts, a front-end, a back-end and a data-warehouse. When choosing which frameworks and libraries to use when implementing, we had to account for two factors, the previous experience and knowledge we had as developers and what Grade used in their tech-stack for their platform. For the front-end of the application the library React was chosen, which was a choice made with support of both factors. Grade uses React for their front-end in their platform and we had previous experience with using the library. For the back-end the .NET framework was chosen. This because we had no major preference of what tool to use and both Grade and the old solution by Peterson and Holm [3] used the .NET framework. Because the data-warehouse already existed we did not have much of a choice when it came to choosing a database server and had to stick with Microsoft SQL Server as the provider.

After the decisions regarding technology was made, the next step was setting up a project to work with. A Visual Studio project was created using the predefined ASP.NET with React template provided in the 2022 version of the editor. This template helped us setup both the front-end and the back-end of the application along

with connecting them with a proxy. The project also makes it easier to publish the implementation because it bundles the front-end and the back-end together into one package were you do not have to deploy each version independently. The decision was also made to run React with TypeScript instead of JavaScript after a discussion with the stakeholders. This mean that the created React code in the project had to be changed to TypeScript. This was done by refactoring the code and changing file extensions and packages. The data-warehouse database from the previous solution was also cloned locally so it would not be altered while creating the new implementation. When the project was created a GitHub repository was setup on Grade's account. This to make it easier for the stakeholders to view the project and for the supervisor to make code reviews. To make it faster to create the front-end and to make some of the components behave more like the rest of the Grade platform, the component library that Grade provides were installed in the project. This library was used through the whole phase and saved some time with the implementation. The overview of the project architecture can be seen in figure 5.2 and the structure of the setup project can be viewed in figure 5.3.

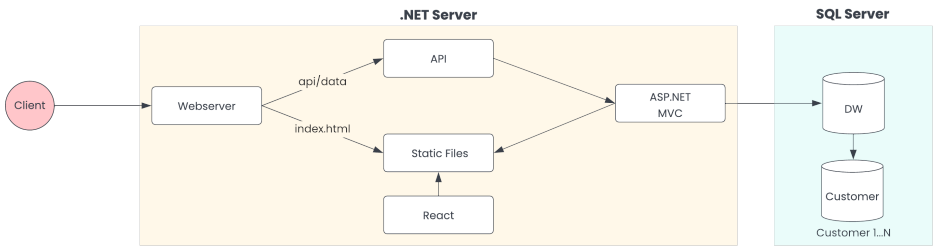


Figure 5.2: The overall technological architecture of the implementation.

5.2 Development

For the whole implementation phase development was done in increments of one to two weeks, where each ending week was adjourned with a meeting with the manager and supervisor at Grade. In the meeting the result of that week was presented and the planning of the upcoming week was performed. This included prioritising features, implementation decisions and more. With each feature developed a code review process were also made were the supervisor review the written code made for the specified feature by commenting on implementation details. This to increase the standard of the code and to make it more similar to the other code of the platform.

Because the project was using the old solution made by Petersson and Holm [3]

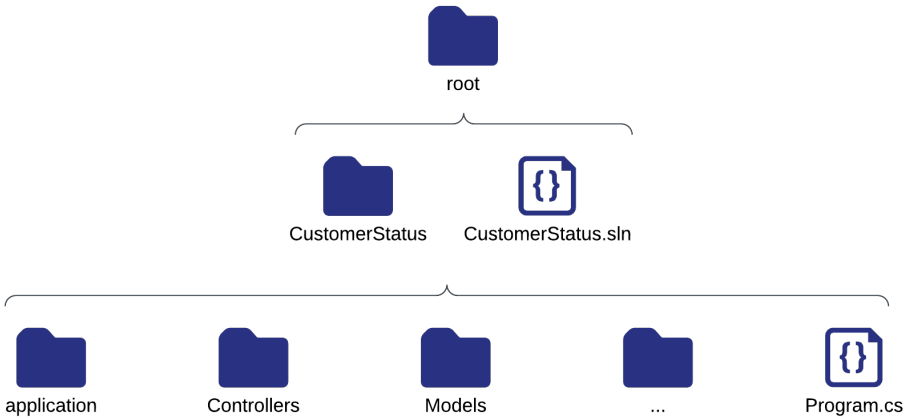


Figure 5.3: The folder structure of the initialized project. The logic for the back-end is located in the controller och models folder while the React front-end is located in the application folder.

the project was limited to the database they had created in their project. The original plan was to extend this solution with more data from the customers. But because of the complexity of the old solution and because there was no-one that was still a employee of Grade had knowledge of the solution this became a lot harder than previously thought. This led to a implementation that was limited to only the React front-end and the ASP.NET API, with the data warehouse layer staying the same as in the previous solution from Petersson and Holm [3]. For the features that needed new data to function properly the interface was still built but mock-data was used instead. Therefor if the data becomes available in the future the implementation has some functionality ready.

5.3 Result

The result of the implementation phase is a fully functional variant of the High-Fi prototype with real customer data from Grade's customers. Because of the high similarity between the High-Fi prototype and the implementation and because the implementation contains some confidential data some views has been excluded. The result of the implementation is shown in figure 5.4 to 5.7. The final implementation was planned to be tested with the same methodology as the High-Fi prototype to be able to compare if the prototype's usability translated into the final product but due to time constraints was something that was excluded from the process.

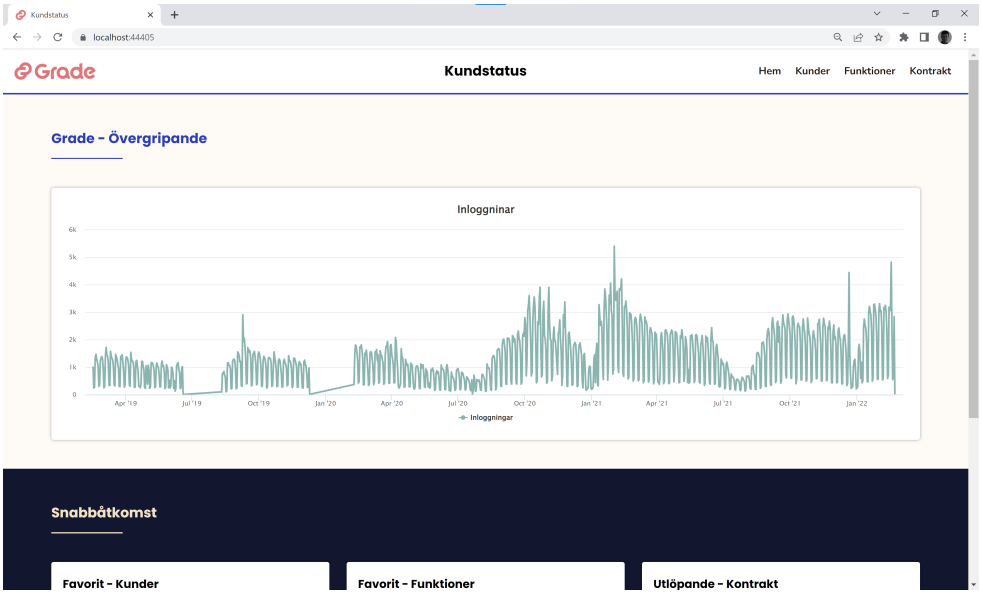


Figure 5.4: The homepage of the final implementation, where the overall data is shown.

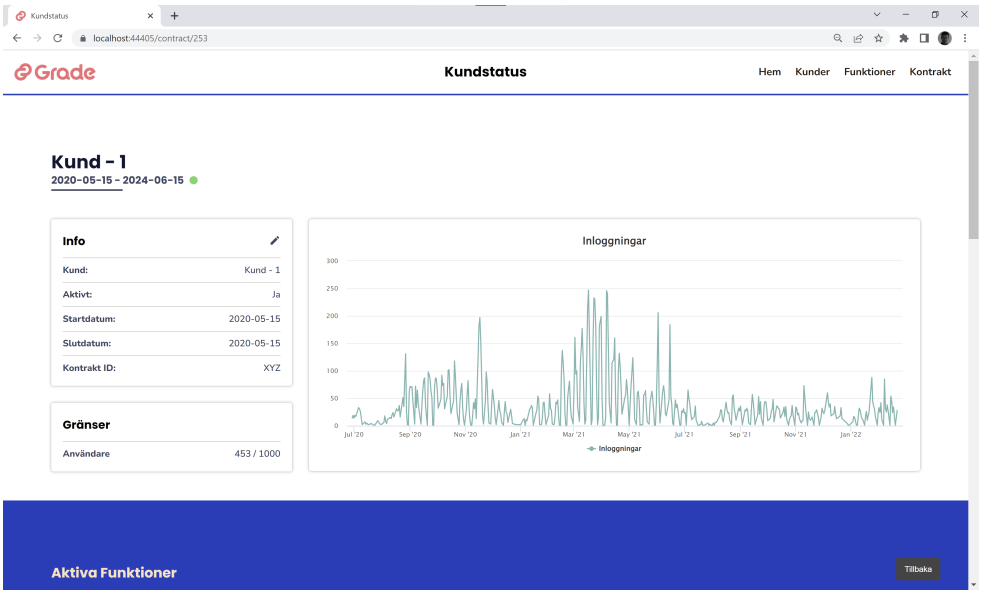


Figure 5.5: The view of a specified contract along with its related data.

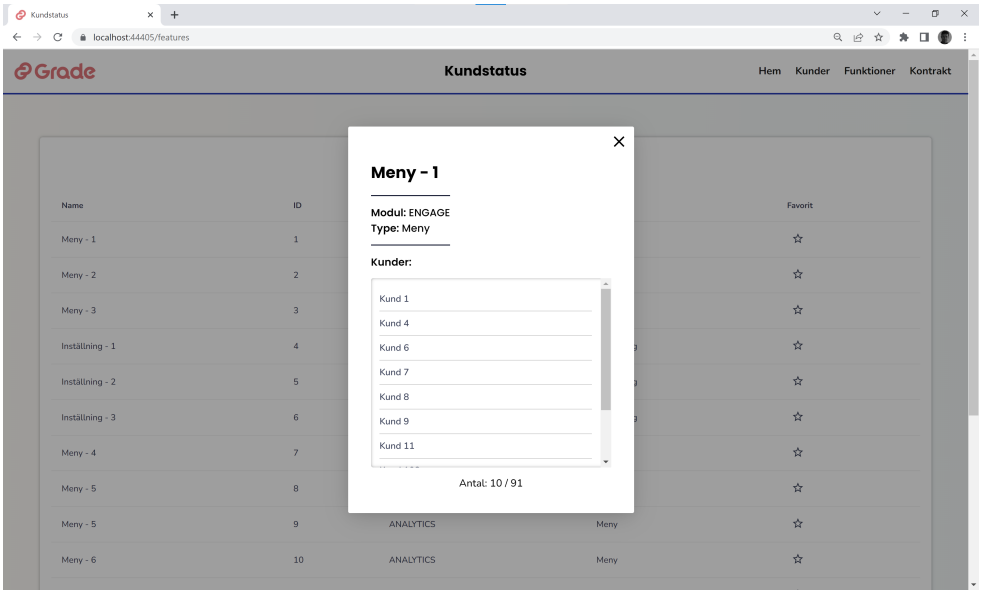


Figure 5.6: The view of a specified feature along with its related data.

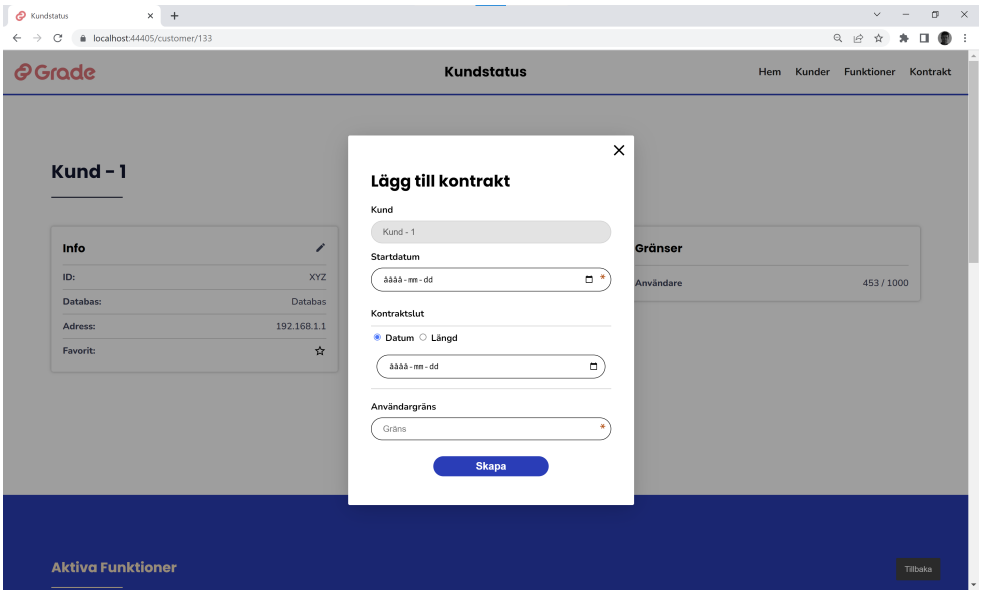


Figure 5.7: The add contract modal for a specified customer.

6 Discussion

In this chapter, the entirety of the project including the process and results will be discussed. The chapter first presents an overall discussion and reflection of the project. Then a discussion of the whole process from investigation to implementation, before moving on to the goals and research questions and ending with the potential for future work.

6.1 Scope

One of the biggest limitation of the project has been the limited size of end users. Because the final product would only be used internally by Grade, there is only around a dozen of users that could benefit from the product. This has made some of the design activities in the project a bit lackluster and some methods has been skipped entirely. The limited target group does not only have disadvantages. With less users it is easier to get feedback from every users which makes it easier to design a solution that suit all. It also removes lots cultural and national differences because of the limited geographical location that the company operates in. However there no doubt that the lack of users to test and evaluate on has affected the usability evaluation and other similar methods.

6.2 Interaction Design Process

One of the major goals in the thesis was to include the interaction design process through the entirety of the project. The process includes of four main activities which will be discussed below.

6.2.1 Establish Requirements

The first activity puts focus on figuring out what the user actually wants from the product. The designer needs to know the requirements the user has of the designed

product. This activity was formed to minimize the lack of end user as best as possible. To get as much data from the limited users as possible, large unstructured interviews was performed to gather as much context as possible. While this generated a lot of great qualitative data, the quantitative data became quite lacking. We also had the opinion that it would be hard to arrive to conclusions if quantitative data of the users would be present because of the limited dataset. However the result of activity still arrived at some concrete data from the users in the form of requirements and personas. In this activity methods such as questionnaire's and workshops could be used, but due to time constraint these were excluded because of previous experience with how much time these activities take, especially workshops.

6.2.2 Design alternatives

The second activity consisted mainly of the conceptual design. The activity resulted in good ideas along with scenarios and a project breakdown structure. However this activity lacked as noticed in the prototyping phase, a method for creating user flows. By creating user flows earlier in the project time could be saved when creating in the prototyping phase. The result of the *design alternatives* activity also served as a good base when trying to agree on a mutual design with the stakeholders of the project. In this activity we wanted to include methods such as brainstorming but as mentioned before due to the lack of designers being available during this period was something that was excluded.

6.2.3 Prototyping

The prototyping activity was split into three parts, Low-Fi, Mid-Fi and High-Fi. Overall the prototyping phase was extensive and probably took more time than it had to. For Low-Fi prototype created with office supplies was probably the most unnecessary time investment of the three. The Low-Fi prototype granted the least amount of feedback from the observations and to much time was invested in the visual design of the prototype. If it was redone, a better approach would be to present quick sketches to the stakeholders and then move on directly to the Mid-Fi prototype. The Mid-Fi prototype was the most time efficient because of how quick it was to setup and the amount of feedback it rewarded as opposed to the Low-Fi prototype. However the Mid-Fi severely lacked design alternatives and by presenting more alternatives, time could be saved in the High-Fi prototype. The High-Fi prototype was also more time efficient than the Low-Fi prototype with regards to feedback, but with most being related to the visual design. The High-Fi prototype also included some design alternatives which was lacking in the other prototypes.

6.2.4 Evaluating

The evaluating activity put the most focus on receiving feedback from the users. All of the designed prototypes went through varying variations of usability testing, which yielded some various results. The Low-Fi and Mid-Fi prototypes were evaluated using only observations with four to five participants. The High-Fi prototype was evaluated using observations, SUS score and task completion rate with four internal users and two external. Ideally these test would be performed with a larger amount of users but because of user limitations this could not be achieved. The internal users gave a high SUS score with an average of 90.6, which according to Sauro [33] is above the 90th percentile and gives a ranking of *A*. However the result need to be viewed with discretion due to the limited set of test users. The internal users also had a high task completion rate, with the external users falling behind. This could be because of the background context needed to use the platform and it shows that it is hard for non end-users to use the platform. Where the activity lacked the most was when it came to evaluating the implementation. This was something that was planned but due to time constraints and the implementation taking longer than previously thought, was something we could not undertake. An evaluation of the implementation could be a good way of seeing if the interactions translated well between prototype and implementation and if the implementation had any flaws with regards to the technology it was developed. Another issue with the activity was that the process lacked something to compare the evaluation to. If the old temporary solution created by Petersson and Holm was evaluated in the same fashion that the High-Fi prototype was, the results could have been compared. In retrospective this was something that could have benefited the project but was not done due to not knowing that the old interface existed during the first phase of the project.

6.3 User-Centered Design

Another goal of the thesis was to use a process that focuses on user-centered design. User-centered design emphasize the use of three main principles which has been followed throughout the project. The first principle *early focus on users and tasks* was one of the main priorities in the first phase of the project. An investigation of the users was conducted where the users was the main focus. This included gathering qualitative data on what ambitions the users had of the solution and whats needs they had. However as mentioned before the principle was hindered because of the limited amount of existing end users. The second principle *empirical measurements* has also been fulfilled to some extent, with data gathered from both the investigation phase and the prototyping phase in activities such as interviews and usability testing. The third and last principle *iterative design* has also been followed to some degree

with iterations appearing throughout the prototyping phase. However as stated in the interaction design process, more iterations could have benefited the project, for example a second iteration of the implementation. The iterations could have also been more well defined, with for example names and numbering. This could have lead to a better comparison between them and make it easier to see what changed.

6.4 Goals & Research Questions

The three research questions has been answered through the whole process of the project. The key points for each research question is compiled below:

How can the interaction design process be used for creating a solution for Grade?

By carefully following and applying the four main activities of interaction design process throughout the entire process a interactive solution could be created that has a strong focus on the users. Activities such as establishing requirements sets clear exception's to the stakeholders and users of what the final product will achieve, while activities such as designing alternatives and prototyping create designs that the user can take part of. The evaluation activity helps to establish how well the solution satisfies the needs and requirements previously specified.

How can the concepts of user-centered design be used to meet the users expectations at Grade?

By constantly including the three principles of user-centered design, it becomes possible to include the users at Grade throughout the whole process. By getting a high SUS score for the High-Fi prototype there were some indications that the users testing the prototype was satisfied with its usability which is some of the users that could benefit from the application in the future. The investigation phase puts a big focus on the first principle of user-centered design, early focus on the user and its task, by investigating who the users are and also including the user in trying to specify requirements for the solution. The prototyping and implementation phase focuses instead on the last two principles, empirical measurements and iterative design. The iterative nature of creating multiple versions of prototypes based on the empirical measurements gathered in each stage, in the form of activities such as usability testing.

How can the implementation of the final design be created?

The implementation of the final design was created using the tools and technologies described in the implementation phase. The implementation became a reality only due to the work that the design process resulted in. Results such as requirements and

prototypes was used to create a implementation that followed the same principles of user-centered design and without it the implementation would have probably looked a lot different.

Goals

The goals of the thesis were partly satisfied, the two first goals *utilize the interaction design process to create the solution* and *utilize the concepts of user-centered design to create the solution* is described in a previous section. The third specified goal of the thesis was to create a High-Fi prototype which was achieved in the prototyping phase. The last goal implement the requirements to make a functioning application,

6.5 Future Work

The solution created in this project has some potential and hopefully will replace the old temporary solution. However there is more aspects that needs to be investigated or developed that did not make the thesis. One of the proposed ideas for future work is to make an evaluation of the temporary solution created by Petersson and Holm to compare the solution from this project with. As mentioned before the final implementation also needs to be properly evaluated. Another potential investigation could be evaluating the user experience of the final implementation. For the development side of the project there is also some work that could be done. A potential avenue for extending the work is to rework or extend the data-warehouse that Petersson and Holm's created [3]. This to include more data that the user wanted from their customers than the current warehouse could provide.

7 Conclusion

This master's thesis aimed to investigate how to design and implement an application for data-driven decision-making, which included designing a interface and develop it into a fully functional application. To achieve this the interaction design process was followed, along with a focus on user-centered design. The result of the thesis is a working application, along with a High-Fi prototype which was used for the design. The thesis describes the process from start to finish with the first phase of the project being the investigation. Here potential users of the application was researched with interviews and then the conceptual design was put forward. The result of the investigation phase was a generalization of the users in the form of personas, scenarios, requirements and a project breakdown structure. Moving into the prototype phase, three different prototypes were created, each with varying degrees of fidelity. For each of the prototypes, the usability was tested to various degrees by using observations on the actual end-users, with the usability testing High-Fi prototype resulting in a high SUS score. After the prototyping phase the thesis moved on to the implementation phase, were the thesis describes the technologies and tools used to implement the application and why they were chosen. The result of the implementation phase is a fully functional and working application that the users can access on the internal network. However due to time constraints he final product did not get to be evaluated. One of the main point discussed in the thesis was that it is hard to do some of the interaction design activities with such a limited user target-group. Another issue is that it is hard to coordinate with users working full time due to the limited time they have for other activities. The thesis also had the difficulty of relying on other projects integrations and that ideas and designs created in the early phase of the project might had to be cut due to the limitations of other solutions. Finally the take-away that following the interaction design process can alleviate implementations, by setting up proper requirements and having prototypes to base the development of.

References

[1] R. David, R. John, and F. G. John. (2021) Worldwide global datasphere forecast, 2021–2025: The world keeps creating more data — now, what do we do with it all?

[2] Grade AB "Grade" grade.com https://www.grade.com (accessed Mar. 9, 2022).

[3] E. Petersson and J. Holm, "Implementation of a Business Intelligence System: a Case Study," M.S. thesis, Dept. of Computer Science, LU, Lund, 2016.

[4] J. D. Gould and C. Lewis, "Designing for usability: Key principles and what designers think," *Commun. ACM*, vol. 28, no. 3, p. 300–311, mar 1985. [Online]. Available: https://doi.org/10.1145/3166.3170

[5] J.-Y. Mao, K. Vredenburg, P. W. Smith, and T. Carey, "The state of user-centered design practice," *Commun. ACM*, vol. 48, no. 3, p. 105–109, mar 2005. [Online]. Available: https://doi.org/10.1145/1047671.1047677

[6] H. Sharp, Y. Rogers, and J. Preece, *Interaction Design: Beyond Human Computer Interaction*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011.

[7] D. Norman, *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, 2013.

[8] *Web Content Accessibility Guidelines (WCAG) 2.0*, web, Web Content Accessibility Guidelines Working Group Recommendation, Dec. 2008. [Online]. Available: http://www.w3.org/TR/WCAG20/

[9] C. Magnusson, K. Rassmus-Gröhn, K. Tollmar, and E. Deaner. (2009) User study guidelines. [Online]. Available: https://www.certec.lth.se/fileadmin/certec/publikationer/HaptiMap_d12.pdf

[10] A. Fontana and J. Frey, "Interviewing: The art of science," in *Handbook of qualitative research*, I. N. Denzin and Y. Lincoln, Eds. Sage Publications, Inc., 1994, p. 361–376.

[11] J. Rubin, *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests*. Wiley India Private Limited, 01 2008.

[12] H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[13] A. Cooper and P. Saffo, *The Inmates Are Running the Asylum*. Macmillan Publishing Co., Inc., 1999.

[14] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '90. New York, NY, USA: Association for Computing Machinery, 1990, p. 249–256. [Online]. Available: https://doi.org/10.1145/97243.97281

[15] J. Nielsen, "10 Usability Heuristics for User Interface Design" nngroup.com https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed Jun. 1, 2022).

[16] J. Brooke, "Sus: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, 11 1995.

[17] J. Lewis and J. Sauro, "Can i leave this one out? the effect of dropping an item from the sus," *Journal of Usability Studies*, vol. 13, pp. 38–46, 11 2017.

[18] J. M. Carroll, "Making use: Scenarios and scenario-based design," in *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, ser. DIS '00. New York, NY, USA: Association for Computing Machinery, 2000, p. 4. [Online]. Available: https://doi.org/10.1145/347642.347652

[19] A. Lester, "12 - work breakdown structures." *Project Management, Planning and Control*, pp. 53 – 59, 2021.

[20] Association For Project Management "Use of product breakdown structures and work breakdown structures" apm.org.uk https://www.apm.org.uk/resources/find-a-resource/use-of-product-breakdown-structures-and-work-breakdown-structures/ (accessed Jun. 3, 2022).

[21] Interaction Design Foundation "What are User Flows?" interaction-design.org https://www.interaction-design.org/literature/topics/user-flows (accessed Jun. 3, 2022).

[22] M. Rettig, "Prototyping for tiny fingers," *Commun. ACM*, vol. 37, no. 4, p. 21–27, apr 1994. [Online]. Available: https://doi.org/10.1145/175276.175288

[23] J. Rudd, K. Stern, and S. Isensee, "Low vs. high-fidelity prototyping debate," *Interactions*, vol. 3, no. 1, p. 76–85, jan 1996. [Online]. Available: https://doi.org/10.1145/223500.223514

[24] D. Engelberg and A. Seffah, “A framework for rapid mid-fidelity prototyping of web sites,” in *Proceedings of the IFIP 17th World Computer Congress - TC13 Stream on Usability: Gaining a Competitive Edge*, 01 2002, pp. 203–215.

[25] S. Lauesen, *Software Requirements: Styles & Techniques*. Addison-Wesley Professional, 2002.

[26] Meta "React" ReactJS.org https://www.reactjs.org (accessed Jun. 7, 2022).

[27] Meta "Components and Props" ReactJS.org https://reactjs.org/docs/components-and-props (accessed Jun. 7, 2022).

[28] Microsoft "What is ASP.NET?" Microsoft.com https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet (accessed Jun. 7, 2022).

[29] Microsoft "What is ASP.NET Core?" Microsoft.com https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core (accessed Jun. 7, 2022).

[30] Microsoft "SQL Server" Microsoft.com https://www.microsoft.com/sv-se/sql-server/ (accessed Jun. 7, 2022).

[31] K. Gorman *et al.*, *Introducing Microsoft SQL server 2019 : Reliability, scalability, and security both on premises and in the cloud*. Packt Publishing, 2020.

[32] Generated Media, Inc. "Faces" generated.photos https://generated.photos/faces (accessed Jun. 21, 2022).

[33] J. Sauro, *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. Measuring Usability LLC, 2011.