The Faceplate Guide

- designing and implementing a digital tool for locksmiths

Emil Hasslöf

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY

2024

BACHELOR THESIS

ROBUST

The Faceplate Guide

Designing and implementing a digital tool for locksmiths

Emil Hasslöf

LUND
UNIVERSITY

The Faceplate Guide

Designing and implementing a digital tool for locksmiths

Published by

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

Subject: Interaction Design (MAML15)
Division: Ergonomics and Aerosol Technology
Supervisor: Elin Olander
Examiner: Kirsten Rassmus-Gröhn

Abstract

This thesis describes the development process of a digital tool that can be used by locksmiths to quickly and easily find the right components for repairing and replacing electronic locking mechanisms. The tool took the form of a mobile application and was developed in collaboration with Robust Ellåsning AB in Malmö, Sweden.

The report describes an iterative, human-centered design process, from data collection and requirement specification, through several iterations of prototype development and usability testing, to a discussion of the project's methodology and results. The Double Diamond-model is used as a framework for the development work and for the structure of the report.

The company's initial requirements for the tool's functionality were verified and supplemented through surveys of users and their work situations. Different solutions were explored in the form of prototypes, which were then used as a basis for usability testing.

The prototypes were built with paper and pen, with the design tool Figma, and finally with the help of React Native, a framework for cross-platform app development.

The problem was also investigated from the company's perspective, as they are also users of the service who will need to update and expand the app's content.

The report concludes with a critical discussion of the project's methodology and results. Usability testing proved to be the most valuable method, both for verifying existing functionality and for generating new ideas and alternative solutions. Usability testing of the final prototype yielded positive feedback, but it remains to be seen how the final product will be perceived by locksmiths around the country.

Keywords: Design process, Double Diamond, Human-centered design, Usability, Mobile application development

Sammanfattning

Detta kandidatarbete berättar om utvecklingsprocessen av ett digitalt verktyg som kan användas av låssmeder för att snabbt och enkelt hitta rätt komponenter vid reparation och byte av elektroniska låsmekanismer. Verktyget tog formen av en mobilapplikation och utvecklades i samarbete med Robust Ellåsning AB i Malmö.

Rapporten beskriver en iterativ designprocess med människan i centrum, från datainsamling och kravställning, genom flera iterationer av prototyputveckling och användbarhetstester, till diskussion av projektets metodik och resultat. Double Diamond-metoden används som ett ramverk för utvecklingsarbetet och för rapportens struktur.

Företagets initiala önskemål om verktygets funktionalitet verifierades och kompletterades genom undersökningar av användarna och deras brukssituation. Olika lösningar utforskades i form av prototyper som sedan användes som underlag för användbarhetstester.

Prototyperna byggdes med papper och penna, med designverktyget Figma och slutligen med hjälp av React Native, ett ramverk för cross-platform apputveckling.

Problemställningen undersöktes även från företagets perspektiv, eftersom de också är användare av tjänsten som kommer att behöva uppdatera och utvidga appens inehåll.

Rapporten avslutas med en kritisk diskussion kring projektets metodik och resultat. Användbarhetstesterna visade sig vara den mest värdefulla metoden, både för verifiering av befintlig funktionalitet, men också för att generera nya idéer och alternativa lösningar. Användbarhetstester på den slutgiltiga prototypen gav positiv feedback, men det återstår att se hur den slutgiltiga produkten uppfattas av låssmeder runt om i landet.

Nyckelord: Design process, Double Diamond, Human-centered design, Usability, Mobile application development

Acknowledgments

I would like to extend my deepest gratitude towards Jan Sjöholm, Omar Budak and everybody at Robust who welcomed me and gave me the freedom and resources to pursue this project at my own pace. Thank you for your patience, your explanations to all my questions and for all the fika.

Thank you to Lars Holmberg for recommending me for this project, helping me get started and keeping me on the right path.

Thank you to Helen, Magnus, Anna, and Lo who supported and inspired me through this journey.

Last but not least, thank you to my supervisor Elin Olander, without whom this report would have been twice as hard to write. I left each one of our meetings full of inspiration and energy.

Lund, March 2024

Emil Hasslöf

Table of contents

1 Introduction

1.1 Background

ROBUST Ellåsning AB is a company based in Malmö, Sweden. They develop and manufacture electronic locks used in public spaces all over Sweden. An essential part of every such lock is the faceplate, a metal plate positioned on the outer edge of the door, which protects the lock and holds it in place. The cylinder, responsible for keeping the door locked and retracting when the handle is pressed down, protrudes through the faceplate. When a lock is installed, the correct faceplate needs to be selected. There are hundreds available. Which one fits depends on many factors such as the type of lock or the dimensions of the door, doorframe or strike plate.

It is not always necessary to use measurements to find the correct faceplate. Sometimes a faceplate from a different manufacturer is already installed and needs to be replaced with a corresponding one from Robust. Translating between faceplates from different manufacturers is possible, but navigating a physical list or table would be impractical due to the large number of options available.

Currently, finding the correct faceplate for any given circumstance can be difficult and time consuming. Robust wishes to make this process quicker and easier through the use of a mobile application. The intended users for such an application are professional locksmiths.

1.2 Purpose

The purpose of this thesis is to design and implement the above-mentioned mobile application. While Robust have expressed some wishes regarding functionality, the complete specification of the application is not determined. Therefore, this thesis aims to examine the problem, and determine which functions and design aspects would lead to the best user experience.

1.3 Constraints

Due to the timeframe, 10 weeks of full-time studies, some of the work I did during my time at Robust is not included in the scope of this thesis. Mainly, I spent several weeks programming a high-fidelity prototype. While the prototype was used for the last round of user tests, the programming work is not proportionally represented in this report.

All interviews and user tests were conducted at Robust's office in Malmö. Traveling to other cities to conduct interviews and tests was not viable due to limited time and budget.

Only three other companies were considered in this project: ASSA Abloy, Safetron and StepLock. This constraint was set by Robust. There are certain standards for electronic locks on the Nordic market, agreed upon by Robust and these three companies. Robust do not wish to acknowledge others not upholding the standards.

Some desired functionality was not viable due to limitations in available data. For example, difference in measurements between faceplates could not be calculated, since measurements of faceplates from companies other than Robust is not available to me.

2 Theory

This chapter gives a brief overview of the theoretical knowledge that was necessary to conduct the project.

2.1 Human-centered design

An interactive product with no users has no value or meaning. Only once the product is being used by real people in the real world, can it create value and benefit for the users, product owners and various stakeholders. No matter how much time and resources were spent on developing the product, it will be worth nothing without users. It therefore stands to reason that the users play an important role in the development of any interactive product or service. In human-centered design, understanding the users, their situation and the problems they face is essential to

creating a solution to those problems that is *usable*. A usable product is one that can help its users to achieve their goals with effectiveness, efficiency and satisfaction.

The following section gives an overview of the six principles of human-centered design described in ISO 9241-210 [1].

1. **The design is based upon an explicit understanding of users, tasks and environments**

All relevant user and stakeholder groups should be identified and considered in the design process. Inappropriate or incomplete understanding of user needs is one of the major reasons for systems failure. The products usability depends on the specified context for which the product was designed. The context includes the users, their goals, the tasks, and the environment.

2. **Users are involved throughout design and development**

User involvement might include actively participating in design, providing relevant data about the context of use, or participating in evaluation of potential solutions. Increased interaction between developers and users leads to more effective user involvement.

3. **The design is driven and refined by user-centered evaluation**

User-centered evaluation is made from the user's perspective. This provides feedback from users, which is essential for any human-centered design process. This type of evaluation can be used to test potential design solutions in real world scenarios.

4. **The process is iterative**

It is extremely unlikely that the initial specification incorporates every detail of the finished product. Iteration is necessary as understanding gradually grows throughout the development process. It is also impossible to incorporate feedback without iteration.

5. **The design addresses the whole user experience**

Making a product pleasant and easy to use is an important part of designing for the user experience, but not the only one. User experience is a very broad term encompassing all the different ways a product can affect the user and those around them. This can be a consequence of the user's prior experiences and personal characteristics, or it can be affected by the environment and the consequences of using the product.

For example, consider the user experience of lighting a lamp on your bedside table. At first glance, one might say that it's defined by how easy it is to find the button and press it, and if it led to the desired outcome, such as providing sufficient lighting to read a book. But perhaps the light, or the sound of the button clicking, could disturb someone sleeping beside you. Maybe the lamp came in a bulky package that

didn’t fit in the garbage disposal. Or perhaps the electrical bill increased and led to financial consequences. These are all factors of the whole user experience.

6. **The design team includes multidisciplinary skills and perspectives**

Fulfilling this principle is not possible or relevant for this project, since the design team includes only one person.

3 Design process

This chapter gives an overview of the design process. It describes the methods, tools and techniques that were used and why they were chosen for this project.

3.1 The Double Diamond

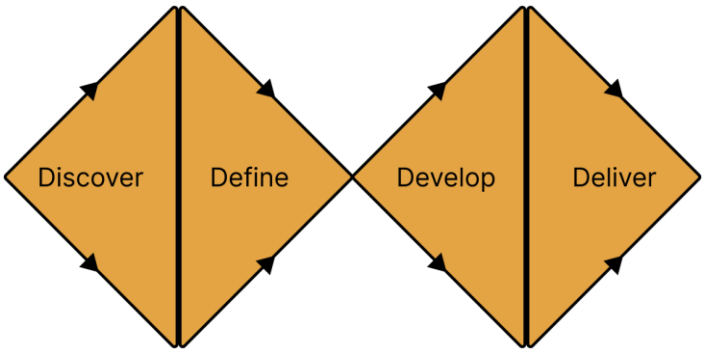


Figure 1. The Double Diamond

An iterative design process can be quite messy and difficult to describe clearly. Instead of describing the process in chronological order, the Double Diamond model [2] is used to enhance the clarity and structure of this report. The design process is represented by four phases as described by the model. The slopes of the diamonds are meant to represent divergent and convergent thinking. Divergent thinking during discovery and development phases enables creativity and innovation. Convergent thinking during define and deliver phases leads to structure and insight.

The four phases, as they are interpreted in this project, can be described as:

- **Discover**: Gather information about the users and the problem they face. Find the functionality that the users want or need.
- **Define**: Define the problem, and requirements for the solution, by evaluating information from the discovery phase.
- **Develop**: Create solutions according to the requirements and current understanding of the problem
- **Deliver**: Test, evaluate and refine the solutions

3.2 Interviews

An interview, according to Preece. et. al. [3, p. 233] "can be though of as a conversation with a purpose". The type of interviews used in this project are referred to as open-ended or unstructured. In this type of interview, the interviewer imposes little control and allows the conversation to flow naturally. While some questions and topics are planned beforehand, the conversation is exploratory in nature, getting sidetracked and going into considerable depth is encouraged.

This style of interview can generate rich and complex data that gives deep understanding of the topic. The interviewee may also mention issues or give insights that the interviewer has not considered. However, the gathered data may be difficult to analyze since it lacks structure and can differ greatly between participants. [3, p. 233-234]

This style of interview is suited to the project because there are only a few users available for interviews, so it is necessary to obtain as much information as possible from each interview, focusing on qualitative data. Gathering and analyzing quantitative data from structured interviews would not be very meaningful for this sample size.

3.3 Brainstorming

The main idea behind brainstorming is to generate many ideas in a small amount of time. This is usually done in a group, as it allows participants to inspire each other. There are many ways to improve the chances of successful brainstorming, such as using props and organizing ideas. The most important principle to keep in mind is that no idea, no matter how silly or strange, should be criticized or debated. [3, p. 370]. Brainstorming was used in this project because it is an effective and natural way to generate ideas.

3.4 Usability testing

The goal of usability testing is to assess whether the intended users can effectively use the product under development to accomplish the tasks it was designed for. A common way to accomplish this is to give the users one or several tasks, and then use a combination of methods to collect data on their performance and experience. Such methods might include observing the users, recording video, logging keystrokes and mouse movements or asking the user to think aloud while working. Some examples of performance measures that can be recorded, given by Preece. et. al. [3, p. 475], are:

- Time to complete a task.
- Number and type of errors per task.
- Number of users making a particular error.
- Number of users completing a task successfully.

Usability testing is a very important method, central to this project and human-centered design overall, since it is an effective way to both gather invaluable feedback from users and let them be involved in the design process. It is also well suited for the iterative approach used, as similar tests can be repeated on new iterations.

3.5 Persona

A Persona is a description of a fictional character that represents users. This character is given a name, a portrait and a description of their personality and characteristics. [4, p. 66] Personas have several potential functions. They can help designers empathize with users [5], they can serve as a representation of gathered data [5] or facilitate communication between project stakeholders [4, p. 66].

While personas are usually built from user-data gathered through interviews and observations [4, p 66], there is no strict definition or mandatory method for developing them.

Since real user data was very limited at the earliest stages of the project, a fictional persona was used to represent the users before real data took its place.

3.6 Feasibility study

A feasibility study, as described by Arvola [4, pp. 17, 42], is the process of exploring what functionality different technical frameworks offer, in order to make decisions about which platform and tools to use.

The feasibility study was conducted to make sure that the chosen platform would support necessary functionality. Starting over because of technical limitations would have been very time-consuming.

A drawback of this method is that the study itself took some time, and was perhaps overly cautions in a project of this scale and limited technical requirements.

3.7 Prototypes

A prototype is an object stakeholders can interact with to explore ideas and concepts before a product is finished. A prototype typically represents some aspects of the product being developed better than others [3, p. 386]. “Fidelity” is a term commonly used to the level of detail of prototypes. A low-fidelity prototype is quick and cheap to produce and is not very detailed. Examples include hand-drawn sketches or storyboards. A high-fidelity prototype is more similar to the final product in visual design, functionality or both [3, pp. 389-392].

A study [7] by Dow, S. P. et al. found that creating and evaluating multiple prototypes in parallel leads to higher quality design, as well as higher designer self-efficacy and confidence, since it’s easier to accept criticism of a prototype if it’s not the only alternative.

Prototyping was essential to the project as usability testing could not have been carried out without prototypes.

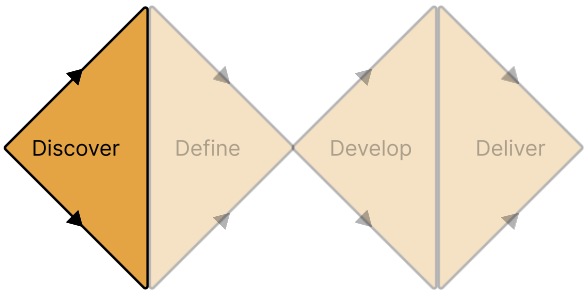
3.8 Overview of activities

	Problem		Solution	
	Discover	Define	Develop	Deliver
Interviews				
Brainstorming				
Usability testing				
Persona				
Feasibility study				
Prototypes				

Table 1. Project activities categorized by the double diamond model

Table 1 gives an overview of where each activity was implemented in the design process as structured according to the double diamond model.

4 Discover



The first phase of the project is focused on understanding the users and the problem. Ideas for design and functionality are generated but not yet evaluated.

4.1 Interviews

Interviews were conducted at Robust's office in Malmö. The participants were five employees at Robust and four visiting locksmiths. The interviews were an ongoing activity through the whole design process, carried out sporadically when participants had some time to spare during working hours. Several of the participants were interviewed many times. The total number of interviews was around 15-20, the exact number was not recorded. The purpose of these interviews was to give better insight into the problem and how it could be solved, and to gain a better understanding of the target users.

Some examples of questions and topics of conversation include:

- What is the purpose of a faceplate, and why are there so many of them?
- How does one decide which faceplate to use?
- What does an average day look like for a locksmith?
- What are the challenges of installing or replacing a lock, and more specifically, the faceplate?
- Describe the environment and situation in which a faceplate is installed.

The interviews yielded qualitative data that was recorded in a journal, organized by date. From this data arose many ideas for functionality and design. Some were direct wishes from users, such as:

- See how many products are in stock
- Search by image
- Prioritize search parameters by importance
- See detailed blueprints
- Translate between any two companies
- Link to order product online

Other ideas were extrapolated from situations and preferences described by users, such as:

- Allow users to send a query by image to Robust, and let them answer which faceplate is appropriate
- Notify Robust that a locksmith is approaching with a query, and provide an ETA
- Make the app usable offline

These ideas were later combined with the ideas from brainstorming, evaluated and turned into requirements, as described in chapter 5.2.

4.2 Brainstorming

Some interviews and conversations with employees at Robust turned into brainstorming sessions. These sessions were not planned ahead of time, but whenever someone came up with ideas I encouraged them to keep going and think of other ways to accomplish their idea or other features related to the topic of conversation. I would also add ideas of my own to further encourage idea generation, making sure not to criticize or even evaluate in any way. All of the ideas from these sessions were recorded together in the same document and saved to be evaluated later. There were around 3 – 5 such sessions in total.

4.3 Role model analysis

The problem of sorting through and finding faceplates is not unique to Robust. There are many web-based solutions that aim to solve this problem in different ways. The solutions studied here are from two of the companies considered in the project. They were studied and compared with respect to the ability to find the correct faceplate, either by search parameters or by a corresponding faceplate from a different company.

The analysis focused on functionality, with the main purpose of gathering inspiration for functions that could be included in the application. The visual design was considered less important, since the application will be optimized only for mobile use and will most likely look very different.

For each website, I tried to translate between faceplates of different manufacturers as well as search for faceplates by different parameters, such as measurements or compatible parts. The necessary steps to accomplish this was noted, as well as additional functionality available to the user while searching for faceplates.

4.3.1 **Safetron**

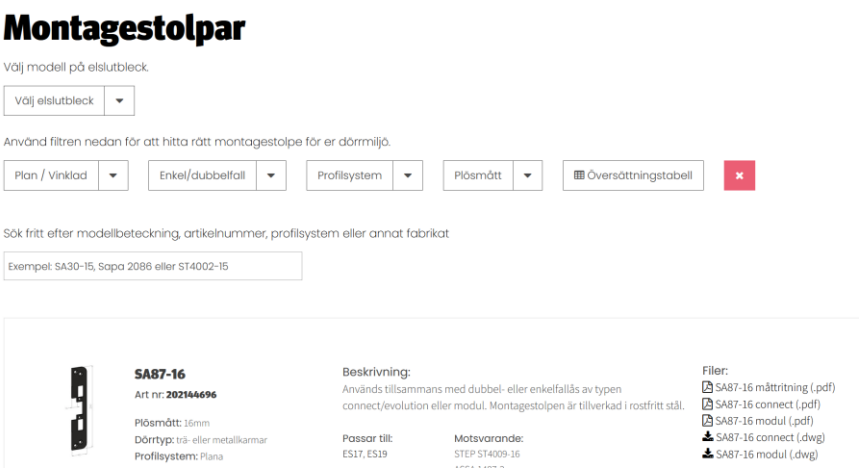


Figure 2. Faceplate search by Safetron, taken from https://www.safetron.com/elslutbleck/montagestolpar

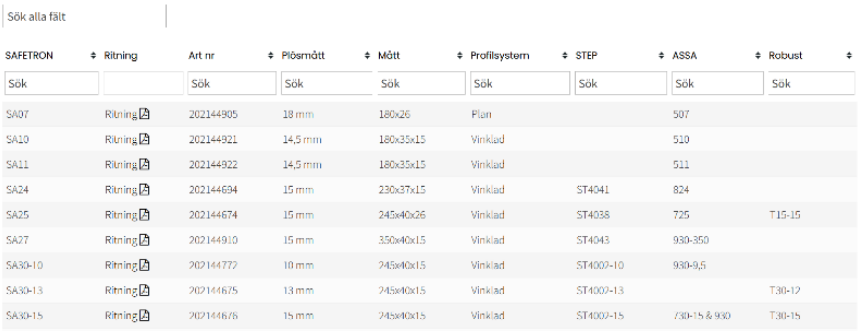


Figure 3. Translation table by Safetron, taken from https://www.safetron.com/oversattningstabell

Safetron provides a product search function depicted in figure 2, and a faceplate translation table depicted in figure 3 on their website. The search function lets the user filter faceplates by five parameters, with the type of strike plate being elevated on its own at the top of the page. Faceplates are presented with simplified blueprints, a brief description of characteristics and compatible parts, as well as links to documents containing detailed technical information.

The translation table gives a succinct overview of the most important metrics, as well as similar faceplates from the different manufacturers. Each column accepts user input. Whenever the user enters any text, the table is instantly updated with data filtered by user input.

While concise and easy to use, the translation table provides only one alternative for each faceplate. This is not entirely truthful, as there are potentially many options

that could fit any given situation. Since there are many columns, the table is hard to fit on a mobile screen with portrait orientation.

4.3.2 **StepLock**

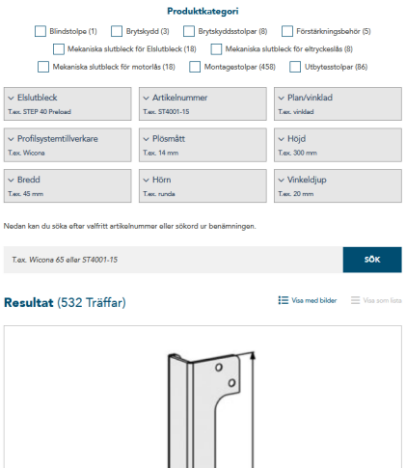


Figure 4. Product Search by StepLock, taken from https://steplock.com/sv/stolpguide/#!?lang=sv&blog_id=1&paged=1&number=50

StepLock have a search function, depicted in figure 4, that combines several product categories, faceplates (montagestolpar in Swedish) among them. The results can be filtered using nine parameters and are displayed in a scrollable list below the input fields. For each result, a simple blueprint is shown that gives a brief overview of the products properties and measurements. Translation between faceplates of different manufacture is not possible.

Compared to Safetron's search function, more search parameters are available. The simple blueprint and description for each faceplate is very similar, but links to more technical information is absent.

4.3.3 **Conclusions**

When presenting the user with faceplate search results, simplified blueprints is an intuitive and elegant way of representing each product. Beyond blueprints, there is a lot of different information that could be more or less important to show.

Translating between faceplates is a task closely related to finding faceplates by search parameters. The filtering mechanism is different, but both use cases lead to the same place: presenting the user with relevant faceplates.

4.4 Persona

A persona, shown in figure 5, was developed to represent the primary user. Personality and behavior were inspired by the first interview that took place, as well as comments and opinions from employees at Robust. However, the persona is fictional and largely based on assumptions. This persona was envisioned as the intended user during the early stages of the project. It was used to aid in making decisions that were not covered by interviews feedback from usability tests. I could ask myself "*what would Ivar think"*, instead of relying on my own personal preferences, which are likely very different from your average locksmith.

Since the persona is not based on actual real data, it was mainly used in the early stages. Changes and modifications to later iterations were driven by feedback from usability testing, thus the persona was no longer necessary.

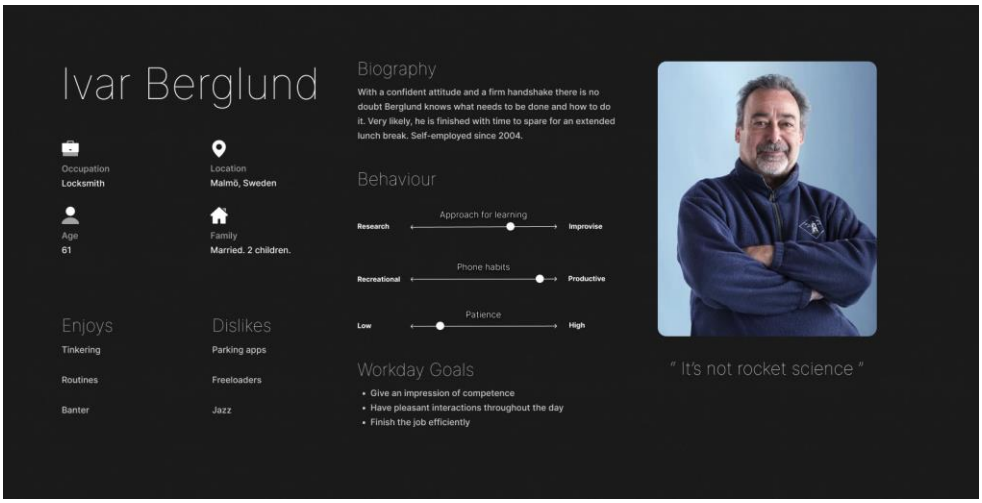
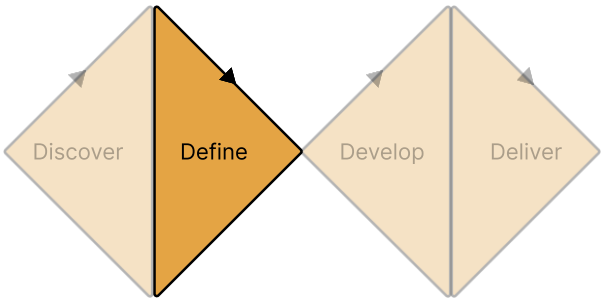


Figure 5. Persona representing a typical user. Photograph by Rod Mclean, used with permission.

5 Define



In the second phase, insight and understanding gained from evaluating data is used to specify requirements, functionality, and design. A technical feasibility study explores which available tools and technical solutions are appropriate for the project.

5.1 Context insights

5.1.1 **Choosing and installing faceplates**

The interviews revealed that faceplates can be deceptively complicated. The process choosing and installing them was repeatedly described to me as a craft requiring skill and experience. Despite hundreds of options, there is often no faceplate that fits the situation perfectly. If the measurements are slightly off or the faceplate is positioned incorrectly, it can cause the door to be too loose and wiggle around, or to not close altogether. Adjusting the position of the faceplate might be difficult or impossible due to existing screw holes. Replacing an old lock with a new one might require removing concrete from inside the door, a laborious process that can damage the door if not done with care.

While there are many measurements and characteristics that affect how well a faceplate fits, some are more important than others. The exact priority is somewhat subjective and each locksmith have their own methods and opinions. However, each person interviewed agreed that “plösmått” is by far the most important one.

Considering these challenges, it is clear that a mobile application will not be able to give a definite answer to which faceplate is most optimal. That decision will be up to the locksmith, and the application can only hope to aid them in making the correct decision.

5.1.2 **Industry and profession**

When asked about the industry and the locksmithing profession, representatives from Robust often use a nostalgic tone. Skilled locksmiths are described to me as "a bit of a dying breed". Over the years, the company has seen the industry change from a careful and deliberate craft where locks are repaired and individual parts replaced, and move towards prioritizing speed, throwing out and replacing functioning components and replacing entire systems because of small problems.

5.2 Design vision

A design vision was formulated in cooperation with Robust:

Trust and respect the locksmith's ability to solve the problem. Help them accomplish this by providing relevant information.

In other words, the app should not attempt to tell the user which faceplate to choose. Rather, it should provide the user with relevant alternatives and metrics to help them make an informed decision.

This vision is grounded in insights about the intended users. As revealed by the interviews and context insights, they often have their own methods and solutions to the same type of problems, developed and refined by years of experience. There is no step-by-step manual or perfect method.

5.3 Requirements

The interviews, brainstorming and understanding of the context of use gave rise to many ideas for functionality. In order to build an interactive prototype in Figma that could be used in usability testing, these ideas had to be prioritized and turned into requirements to be implemented in the prototype. Since no user feedback was available at this time, the following first requirements were formulated according to my own intuition for what was most likely to be important, as well as discussion with employees at Robust.

- **Translate faceplates from StepLock, ASSA or Safetron to their corresponding variant from Robust.** This is the core function of the application and the main user need to address.
- **Display several faceplates at once with name and blueprint.** This goes in line with the design vision of providing information that lets the user solve the problem.
- **Reach results with as few steps as possible.** The application is a tool that should be quick and efficient to use by an experienced user.
- **Find faceplates either through translation or search by parameters.** If no faceplate is available to translate, the user needs to find the faceplate by other means.
- **Filter faceplates by multiple parameters simultaneously.** The validity of this functionality was confirmed by observing a locksmith as he demonstrated a similar tool during an interview.
- **Give Robust full control over the data in the application.** Another user group is Robust themselves, and their user experience is also important. The application needs to be integrated with their existing technology and methods.

After the first round of usability testing, using the prototype built in Figma, the following requirements were added. The testing process is described in chapter 7.

- **Animated toggle between translation and search modes.** This gives the user feedback and makes it easier to understand what happened.
- **Show each faceplate as distinct item with a surrounding border.** One user wished to make the faceplates stand out more.
- **Show available user inputs with a clickable dropdown.** This is necessary to let the user know which alternatives are available. It also makes the app quicker to use.
- **A button to clear input fields.** After observing users erase one character at a time, it became clear that this would improve the experience.
- **Sliding animation for search results.** This is a way to provide the user with feedback that the input was recorded and search results have been updated.

After the second round of usability testing, using the application prototype running on a smartphone, the following requirements were added.

- **Label each faceplate with the search input that caused the match.** Incomplete search input that could be one of several alternatives show results for all those alternatives, i.e. ST65 could be ST6503, ST6519 or ST6577. This is sometimes beneficial, but requires feedback so the user understands the result.

- **Search for a specific strike plate and show results for the series it is a part of.** This feature was requested by a locksmith during testing.
- **Standardize input formatting.** After the user finished entering the input, it could be formatted according to a standard. This would eliminate worry that the formatting is incorrect.

5.4 Technical feasibility study

A technical feasibility study was conducted in order to determine which technologies and frameworks will be used for cross-platform development, as well as how the application can fetch and update the faceplate translation table and product data.

5.4.1 **Cross platform development**

The application should behave, if not identically, at least very similarly, on a wide range of Android and iOS devices. There are many tools and frameworks that allow for developing cross-platform applications from a single codebase. Using such technologies will not only decrease development time, but also help ensure consistency across platforms. Among the most used are React Native, Flutter and Ionic [8]. All three have a wide range of features that cover all the functionality needed for this relatively simple application, including hot reloading (the ability to instantly see changes to the code while testing on a physical device), which is a great convenience that can speed up the development process significantly.

React Native was chosen mainly because of my own desire to familiarize myself with the React framework, due to its immense popularity in web development.

5.4.2 **Retrieving faceplate translation table**

In parallel to this project, Robust is building a translation table that lists corresponding faceplates from different manufacturers. The data is presented as a table on their website, as shown in figure 6. The table is built using a WordPress plugin that automatically generates HTML, CSS and JavaScript and handles all interaction with the database. With this solution, Robust can enter and update data without having to do any programming.

Plösmått | Profil | Höjd | Bredd | Passar elslutbleck | ASSA | Safetron | STEP | Återställ | Sök: st4

Måttritning	Robust	Plösmått	Profil	Höjd	Bredd	Passar elslutbleck	ASSA	Safetron	STEP
	T18	20 mm	Plana stolpar	245	45	SERIE 200	731-20, 931-20, 933-20	SA31-20	ST4001-20
	T69	24,5 mm	Plana stolpar, SCHÜCO profil					SA90	ST4052
	T74	20 mm	Plana stolpar, SAPA profil, SAPA 2074				9487-10	SA43	ST4023
	T88	23 mm	Aliplast profil, Plana stolpar, Purso profil, Purso LK78H, Wicona 65	245	44,5	SERIE 200	946-W		ST4035, ST6503, ST6577

Figure 6. Faceplate translation table, from https://robust-se.com/translate-table/

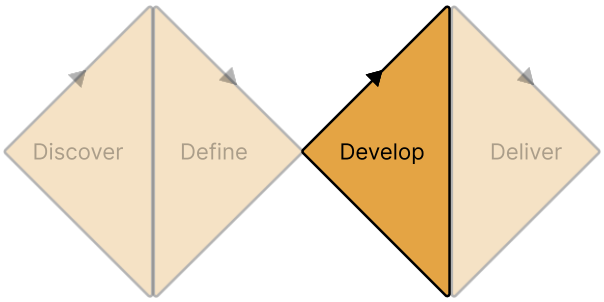
In order to use this information in the application, the data needs to be retrieved either from the website or directly from the database. Both options were explored.

Direct access to the database proved difficult due to how the data was organized by the various WordPress plugins used by Robust. Furthermore, granting the application access to the database would require authorization and could potentially pose a security threat if not done carefully. As an alternative, the option to scrape data directly from the website using HTML parsing was tested.

As a simple proof-of-concept, a script was built with JavaScript, running locally using the node.js runtime environment. The script collects data from the website using the JavaScript fetch() function. The response is raw HTML, which is then parsed using the free open-source library *Fast HTML Parser* [9]. The relevant data can then be extracted and encoded into JavaScript objects, ready for use in the application. A similar solution could easily be implemented in the application.

This proof-of-concept showed that fetching the data this way is viable. Unless unexpected errors arise, this approach should be sufficient for the purposes of the application being developed. The biggest advantage to using HTML parsing is that Robust can modify the source of the data through WordPress, without having to update the application in any way, and with no additional configuration needed.

6 Develop



In the third phase of the project, potential solutions are developed which can later be evaluated through usability testing.

6.1 Prototypes

Three different types of prototypes were developed during the project; hand-drawn sketches, digital prototypes built with Figma, and a mobile application developed with JavaScript and React Native. This section describes how the prototypes were built, and their role and purpose in the project.

6.1.1 **Prototype purpose**

Using a model proposed by Houde & Hill, prototypes can be categorized by their purpose and what they are trying to convey, rather than attributes such as used materials, functionality or fidelity. Focusing on the purpose of the prototype and the design artifact it represents can help us to think and communicate about the design, rather than the prototype itself. [10]

A prototype can be categorized by placing it into a triangular, where the corners represent three design aspects.

Role refers to how the product is useful, which function it would serve in the user's life.

Look and feel refers to how the product looks and how it feels to use. **Implementation** refers to details of how the product is built, the "nuts and bolts" as Houde & Hill put it. A prototype can be placed into a diagram to show which of these aspects the prototype is intended to represent.

Using this model, figure 7 gives an overview of what the different types of prototypes are meant to convey.

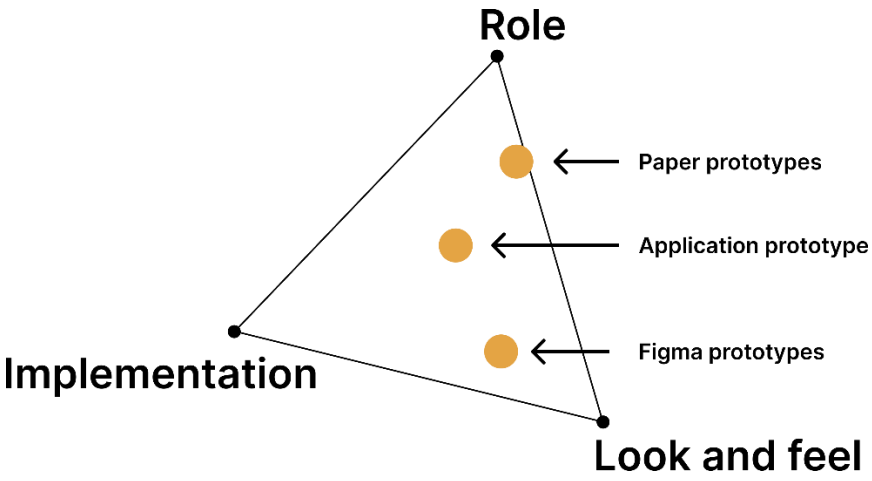


Figure 7. Prototypes categorized by purpose in Houde and Hill's model [10]

6.1.2 **Hand-drawn sketches**

As a way to explore and generate ideas, simple sketches were drawn on paper. According to the benefits of parallel prototyping [7], many prototypes were developed simultaneously. The first sketches were inspired by the initial problem formulation from Robust and the role model analysis. Further versions were created by iterating and making alterations. These low-fi prototypes were mainly used to communicate ideas and facilitate brainstorming during interviews. While giving hints of potential visual design, their main purpose was to convey ideas about functionality. Examples of these sketches are shown in figures 8 and 9.

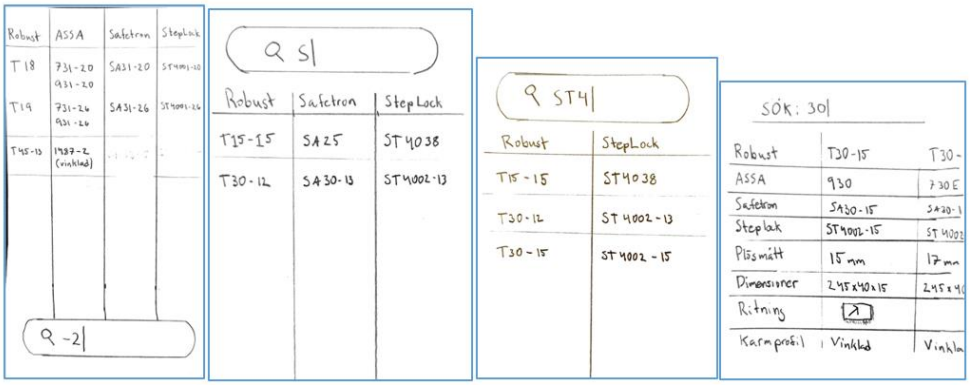


Figure 8. Examples of prototypes exploring table format.

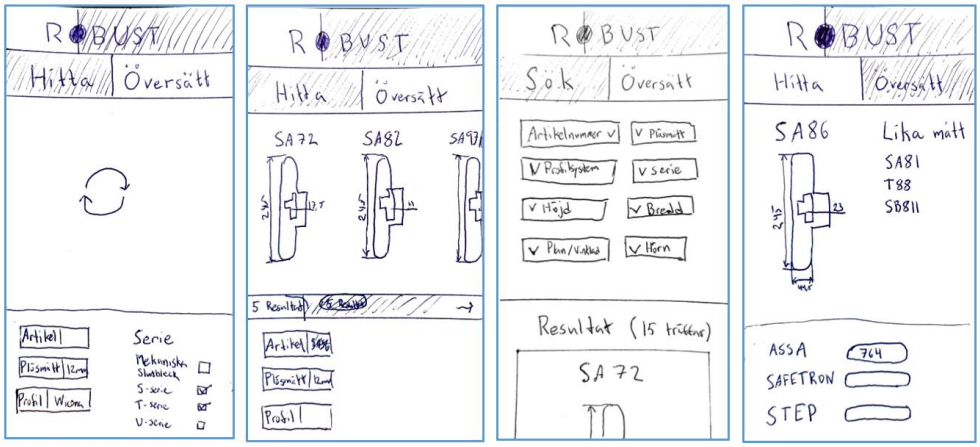


Figure 9. Examples of prototypes exploring blueprints.

6.1.3 **Figma**

Digital prototypes were created using Figma, a web-based program commonly used for design and prototyping. By copying and pasting while changing small details, very quick iteration and parallel prototyping was possible. Some of the prototypes created this way are shown in figure 10.

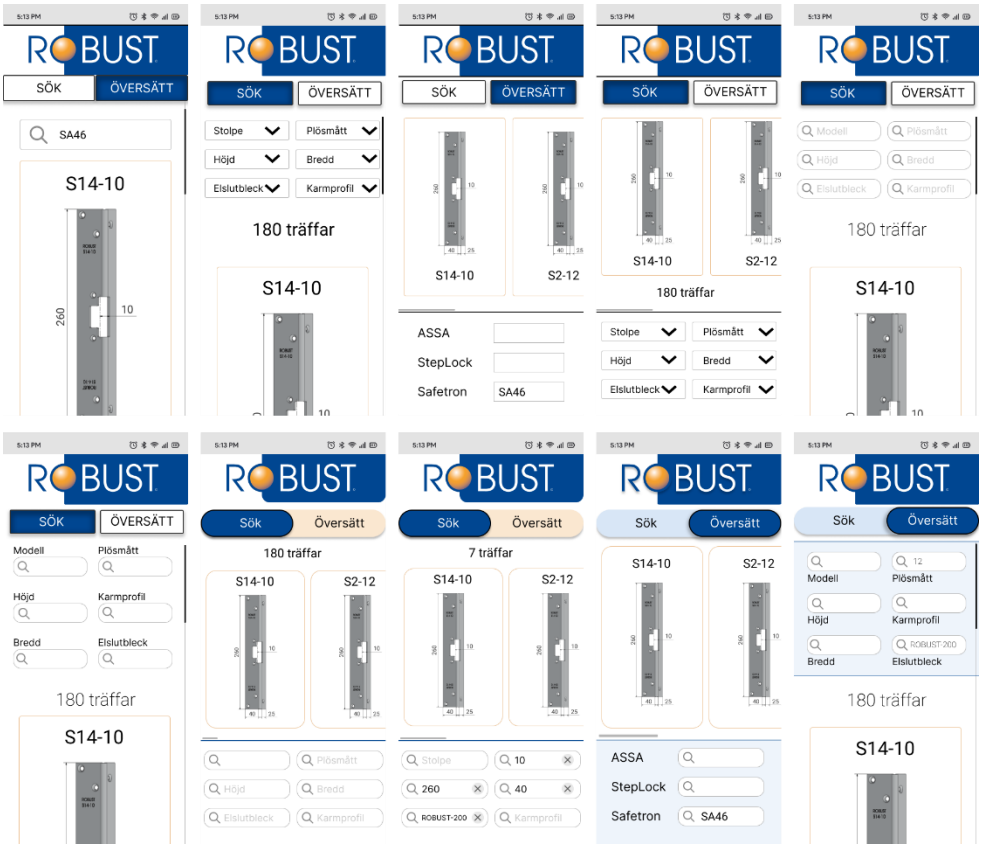


Figure 10. Iterations of digital prototypes

With the functionality mostly decided, these prototypes were more focused exploring and refining the look and feel of the application. Visual fidelity is much higher compared to the paper prototypes.

Figma supports flexible automatic layout of components with padding and margins, similar to CSS flexbox or style props in React Native. This functionality gave hints about how the application could be implemented and made the transition to the JavaScript application smoother.

Figma also lets the user create interactive prototypes, where parts of the screen are clickable and lead to other screens. The prototype can then be run on a smartphone using the Figma app. The prototypes used in the first round of usability tests were built this way.

6.1.4 **Mobile application prototype**

The final prototype was created using React Native and Javascript. The details of the programming work are not described since they are not within the scope of this

thesis. Figure 11 consists of screenshots from this final prototype, running on an Android device.

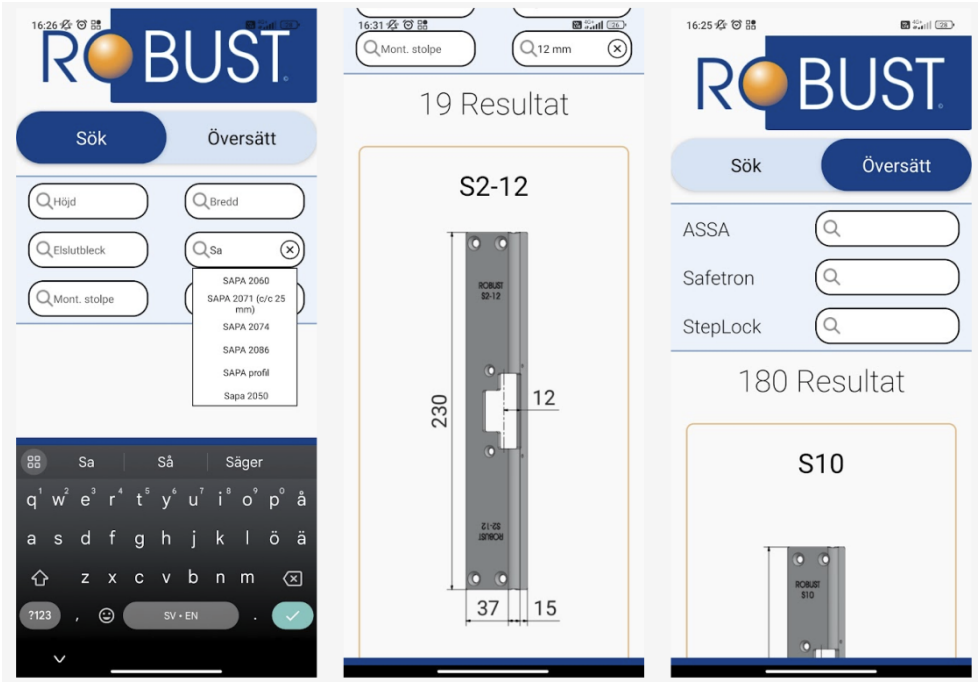
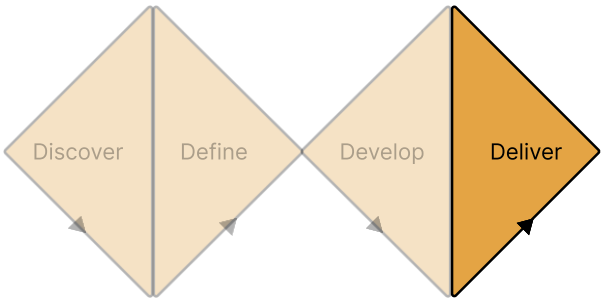


Figure 11. Screenshots from the final prototype

The mobile application prototype has two main screens. One for searching, and one for translating. From the search screen, the user can input several parameters at once, then scroll down to find faceplates matching all the parameters they entered. Parameters can be typed in manually or chosen via a dropdown list. If the user starts manually typing, the dropdown list only contains options matching what the user already typed. If one or more other parameters are already in place, the dropdown list contains only the options that would result in at least one matching result.

The translate screen lets the user enter faceplates from one of three companies other than Robust, and then scroll down to find faceplates from Robust that match the entered faceplate. The dropdown list options do not guarantee matching plates like the ones in search mode. They always show all options, even if they would result in zero matches. Adapting dropdown options would let the user indirectly translate between the other manufacturers via the dropdown lists. Robust do not wish to allow this.

7 Deliver



In the fourth and final phase, solutions are evaluated, tested and refined.

7.1 Refining graphic design

Using my own digital prototypes as a starting point, the graphic design was refined together with a co-designer [11]. By iterating quickly in Figma, many options could be explored and discussed. Some of the design rationale is discussed below, and the final result is shown in figure 12.

7.1.1 **Affordances**

Contrasting color and shape together with visual depth from drop shadows give the impression that the top bar can move and be toggled. Showing only part of the faceplate makes it clear that there is more to be seen. Together with the scroll bar, it gives a clear signal to the user that they can scroll down to see more. These affordances were later confirmed by observations during usability testing, as every user intuitively clicked the top bar to toggle between modes, and all but one instantly scrolled down to see the results.

7.1.2 **Colors**

The color scheme was taken from the company logotype to provide consistency, with some lighter shades of blue used for contrast between elements. A dark-blue bar was added to the bottom to provide better visual balance [11].

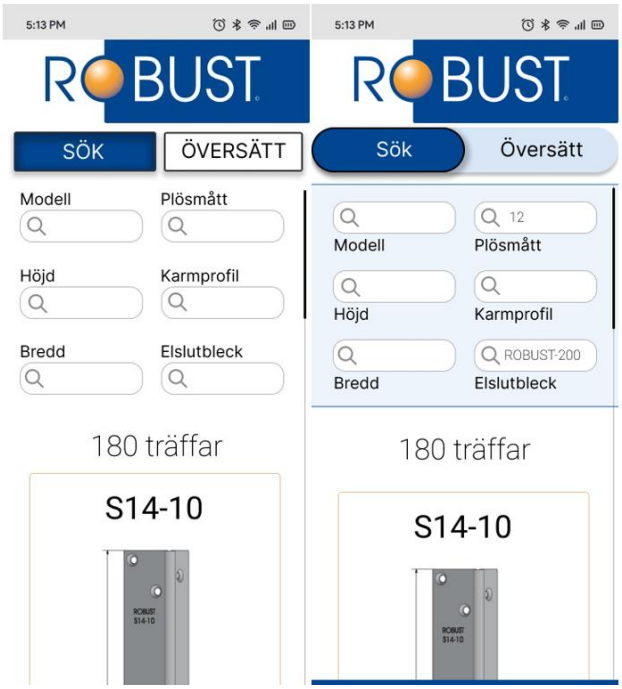


Figure 12. Before and after refining design

7.2 Usability testing – Figma

The first round of usability testing was done using an interactive prototype built in Figma. Six users were given a smartphone with the prototype running and showing the faceplate translation screen. They were given the task of finding which faceplates have a “plösmått” of 12 mm and no additional information. To accomplish their task, users would have to first toggle to “search mode”, click the input field for “plösmått”, select “12mm” and then scroll down through the list of options.

Since only one locksmith was available for testing at the time, convenience sampling [3, p. 228] was used to gather more data.

Six tests were conducted with six different users. Four users completed the task successfully and quickly with no errors.

One user completed the task but did not intuitively scroll down to see all the results. The user claimed that the reason they did not scroll was that they didn't realize anything had changed.

One user toggled back to translation after entering the "plösmått". They reasoned that they would have to press the "Translate button" for anything to happen.

Both failures could potentially be a consequence of lacking feedback that the input was recorded and search results were updated.

7.3 Usability testing – Application

Two locksmiths were available to test the application. Since the tests were so few, they were focused on qualitative data. Two tests were conducted in an unstructured manner where users were encouraged to explore the application and share their thoughts. Some tasks and instructions were given to progress the test, but each user was free to explore the application how they saw fit.

7.3.1 **Test 1**

In the first test, a user was given a smartphone with the application running and asked to translate from a specific faceplate, ST6577. At first, they did not toggle from search to translate, but rather entered the faceplate serial number in the wrong field and got no matching results. After explaining that the application has one mode for translating and one mode for searching, the user immediately found the toggle, started typing the serial number and then pressed the correct alternative in the dropdown that showed up.

When trying a second time, the user entered ST65. When the dropdown showed three alternatives; ST6503, ST6519 and ST6577, the user pressed enter. This revealed an oversight in the application where faceplates matching all three alternatives were shown.

The user requested a feature to search for faceplates that fit a specific strike plate. Each strike plate is part of a series where all strike plates in that series fit the same faceplates. Only the series is searchable, not the specific strike plate, since they all fit the same faceplates. However, if a user does not know that the strike plate they have is part of a series, they might wish to search for that strike plate specifically.

When entering search parameters that yielded no results, the user expressed worry that they might have entered the information with incorrect format, such as character case or lacking symbols.

The user had no problem recognizing all the different parameters and what they were for. The interaction via clicking, typing and scrolling worked as expected. They said their overall impression was good and they though the application would be useful. They claimed that they would definitely use it in their work.

7.3.2 **Test 2**

The second test revealed the same flaw as the first one. The user entered an incomplete name for a faceplate and pressed the enter key, thus getting results matching two different faceplates. Apart from this, the user was impressed by the application and found it efficient, convenient and useful. They had no trouble navigating the interface. Clicking, scrolling and typing was all done correctly and intuitively. They said that it might be useful to have a more detailed blueprint available, but that it would be used very rarely. The current information about each faceplate would be sufficient in almost every case.

8 Discussion

This chapter discusses some common design principles in relation to this project. Further, the project methodology and result are critically examined. Last comes a conclusion summarizing the chapter.

8.1 Design principles

Preece et. al. describe some of the most common design principles that can be helpful to keep in mind when designing for the user experience. These principles are abstract and generalizable, not a detailed instruction, but rather a form of dos and don’ts of which features to include and what to avoid.

This section discusses the final prototype in relation to each of these common principles. It highlights how the principles were taken into account in the design, and how they could be used to guide further improvement.

8.1.1 **Visibility**

The application is split into two modes in order to make all functions visible on the limited screen space of a smartphone. As the user scrolls down through the alternatives, the input menu scrolls out of view, prioritizing the visibility of the results.

When scrolling through the results, each faceplate with its blueprint takes up almost the entire screen, making details easy to see, but it is difficult to compare results to each other. Further testing could decide if this is a good trade-off.

8.1.2 **Feedback**

Feedback means giving the user a clear indication that their actions have impact. Not knowing if something happened due to lack of feedback can lead to a very frustrating experience. The importance of feedback was clearly demonstrated in several user tests on early prototypes. Users would enter search parameters and the result would update instantly, but this was not easy to discover without scrolling down first. This led to the user thinking nothing had happened. This issue is improved in the final prototype by making the result area blank while the user is typing into the search fields, making it more obvious that the app is reacting to the input. There is room for improvement though, as the blank area does nothing to indicate that the results are actually updated.

8.1.3 **Constraints**

Constraints can be used to restrict the user from certain interactions. There are currently no constraints in the application. Restricting options from a dropdown menu that would lead to zero search results found could perhaps make the searching experience faster and more intuitive.

8.1.4 **Consistency**

Through consistency, the designer can give the user hints of which elements have similar functionality. The input fields all look the same, but their appearance is slightly altered when switching between the two modes, making it clearer that the

functionality is a bit different. The theme of the application aims to be consistent with the company logo and website.

8.2 Methodology

8.2.1 **Persona**

The persona method was chosen for this project mainly because of the limited interaction with real users during the early stages. As more user interaction took place, the persona became less relevant and was eventually replaced by real user data. While a persona based largely on assumptions is of course not optimal, it can still facilitate empathy, as well as mitigate bias towards my own personal preference. There are cases where design companies have built personas based entirely on assumptions and still reported high satisfaction with the method [6]. Human-centered design requires a user to design for, and even if that user does not align perfectly with reality, it is still better than no user at all.

8.2.2 **Interview bias**

The interviews all took place at Robust's offices in Malmö. This means all interviewed locksmiths operate in close proximity to the company. This changes their perspective on the company compared to users in the rest of the country. They are more likely to have a personal relationship with the company and perhaps be less critical of new ideas. Locksmiths that operate in Malmö can also quickly and easily visit Robust to get information on their products and collect hardware directly from the shelves. This means that the app is perhaps more important for users not close to Malmö. Getting feedback from these far away users would likely be more valuable.

8.2.3 **Working efficiently as a solo designer on a small project**

While searching through literature and planning my methodology, many of the methods I found were described as being used in large projects with multiple designers. In particular, Arvola gives a very thorough description of what he calls the "concept phase" [4, pp. 39-84], similar to the *Discovery* of the double diamond. Adapting Arvola's descriptions of design methodology to fit my much smaller project with only one designer was a challenge.

With the clarity of hindsight, my estimation is that much of the planning I did before developing the first prototypes were time-consuming and perhaps not worth the

effort. In particular, developing a persona to represent users and doing a technical feasibility study was time that could have been better spent building and testing prototypes.

One argument for doing thorough research before development begins is that mistakes become more expensive to fix later on. In this case, however, very little time was spent making changes, and the product is small and simple enough that making extensive changes later on would have been viable.

Another argument would be that once development starts, it is possible that improvements lead only to a local maxima. Perhaps the best solution would require rethinking the product from the ground up in a way that cannot be found by testing existing prototypes. I would argue that in this case, finding radically different approaches was easier to do through parallel prototyping, focusing on user tests over user research. I think it is likely that this conclusion would have been different if the project was larger and more complex.

The user testing was very successful, leading to validation of existing ideas as well as many ideas for improvements and new functionality. If I were to start a similar project today, I would aim to develop many different prototypes as early as possible and focus a lot on usability testing.

8.2.4 **Unexplored ideas**

Due to the limited timeframe of the project, many ideas from the discovery phase were left unexplored. While formulating requirements for the Figma prototype used in the first user tests, choosing which ideas to pursue and which ones to leave for later or abandon were done through discussion with employees at Robust and according to my own intuition. With more time and resources, these ideas could have been evaluated more thoroughly, either through more prototypes or better techniques for data analysis. It is very possible that some of them could have led to a better user experience in the end.

8.3 Result

The final prototype has received good feedback from both users and Robust. The two locksmiths that did the final test both agreed that it would be useful in their work, which is promising. However, since the application is not yet released on the public stores it is difficult to draw final conclusions. Time will tell if the application is adopted among locksmiths in the rest of the country.

The way for Robust to enter new information to be used in the app is working so far, but it is not very future proof since its entirely dependent on the company

website. If Robust should decide to move or redesign their website, it could cause the app to break. A better solution to this problem would make the application more usable from the company's point of view.

8.4 Conclusion

Studying the context, along with Robust's initial specification and collaborative brainstorming yielded many ideas for how the interaction between user and application should be designed.

Many of these ideas were implemented and tested throughout the design process, while others remain as potential improvements to be explored in further iterations.

Usability testing proved to be a very valuable source of feedback. Many improvements came as a direct result of user feedback from the tests.

While I was working as a solo designer and developer, the final prototype is the result of collaboration between me, the employees at Robust and 4-5 locksmiths that regularly visit their offices.

Using the React Native framework, the application runs natively on both android and iOS smartphones.

References

[1] International Organization for Standardization (ISO). (2010) *Ergonomics of human-system interaction – Human centered design for interactive systems* (ISO 9241-210)

[2] The Design Council. The Double Diamond. Retrieved November 2, 2023, from https://www.designcouncil.org.uk/our-resources/the-double-diamond/

[3] Preece, J., Rogers, Y., & Sharp, H. (2015) *INTERACTION DESIGN – beyond human-computer interaction.* John Wiley & Sons Ltd

[4] Arvola, M. (2014). *Interaktionsdesign och UX – om att skapa en god användarupplevelse.* Lund, Sweden: Studentlitteratur

[5] Nielsen L. Engaging Personas and Narrative Scenarios. Copenhagen. 2004.

[6] Nielsen, L., & Storgaard Hansen, K. (2014, April). Personas is applicable: a study on the use of personas in Denmark. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1665-1674).

[7] Dow, S. P. et al. Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-Efficacy. *ACM Transactions on Computer-Human Interaction*, Vol. 17, No. 4, Article 18. December 2010

[8] Statista. Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2022. Retrieved November 1, 2023, from https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/

[9] NPM. Fast HTML Parser. Retrieved November 2, 2023, from https://www.npmjs.com/package/node-html-parser

[10] Houde, S., & Hill, C. (1997). What do prototypes prototype?. In *Handbook of human-computer interaction* (pp. 367-381). North-Holland.

[11] Anna Hasslöf, Graphics designer and Architect, examined from Lunds Tekniska Högskola, Lund University, Sweden. Personal conversation (2023, 10, November)