CHALMERS

ResursRätt

HÄLSOSAM MAT TILL ALLA

User-focused App Development for Räddningsmissionen

Application for donations from companies

Bachelor's Thesis within TIDSL and TIDAL

JOAKIM DAHLSTRÖM AND STORM ROTHOFF

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK

CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2025
www.chalmers.se

Bachelor's Thesis 2025

User-focused App Development for Räddningsmissionen

Application for donations from companies

JOAKIM DAHLSTRÖM AND STORM ROTHOFF

CHALMERS

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola
Göteborg 2025

User-focused App Development for Räddningsmissionen
Application for donations from companies
JOAKIM DAHLSTRÖM AND STORM ROTHOFF

Supervisor: Mario Caneiro, Instituionen för Data- och Informationsteknik
Examiner: Nicholas Smallbone, Institutionen för Data- och Informationsteknik

Bachelor's Thesis 2025
Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Cover Art: Logo for the application "ResursRätt".

Skriven i LaTeX
Göteborg 2025

User-focused App Development for Räddningsmissionen
Application for donations from companies
JOAKIM DAHLSTRÖM AND STORM ROTHOFF
Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola

Abstract

The project is revolved around an mobile app called “ResursRätt” by Räddningsmissionen. The app allows companies to donate residual products that Räddningsmissionen then sells for a low price in their stores. The app collects information on what is donated and how the delivery of said products will be done. Räddningsmissionen either picks up the delivery or the companies delivery it to their warehouse.
The goal of the project is to improve the user experience of the app, this is done through a user study where different potential users are interview. Users also got to test the original app and express their opinions of it. The users inputs where then used in creating different concepts and prototypes.
The prototypes where made in Figma before moving on to coding the final version of the new app. The users also got to try the prototype and give inputs that where implemented in final version. The final result of the project landed in a more improved upon UI that is more efficient and practical than the original version.

Keywords: UX, App, UI, Functions, Structure, Users.

Acknowledgements

We would like to thank Emelie Heilmann and Johanna Älisgård at Räddningsmissionen for letting us work on this interesting project. They have provided helpful feedback and been positive to our changes. We would also like to thank Erik Ulfung Arnbom, Zebastian Björkqvist, Jean Paul Khalaf, Josefin Wallberg, and Jonathan Rootzén which created the original code. They have been a lot of help in understanding the structure and how to run the code. We would like to thank everyone that gave feedback and participated in the user study, without their thoughts we would not have been able to know what was needed to be improved. Finally we would like to thank our supervisor Mario Caneiro for having weekly meetings with us and for the guidance he provided.

Joakim Dahlström and Storm Rothoff, Göteborg, June 2025

Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

Cla	Clarity
Com	Complexity
Con	Contintuiy
CW	Customer Worth
E	Effieciency
I	Intuitive
Int	Integration
MF	Main Function
MMF	Meeting Main Function
MSF	Meeting Sub Function
R	Result
SUBF	Sub Function
SUPPF	Support Function
T	Total
UI	User Interface
UX	User Experience
W	Weight

Contents

1
Introduction

1.1 Background

This project is about evaluating and improving Räddningsmissionens current app “ResursRätt". The app “ResursRätt" is a tool for companies to fast and easily donate various residual products such as food, clothes and other wares. When a company has registered a donation through the app Räddningsmissionen collects the items and stores them to their warehouse, where they are later sold or donated at one of their locations.
The project will primarily focus on the user experience from the various parties that use the app, where both the company donating and users from Räddningsmissionen are included. This involves improvments in information handling and presentation in the app, and also exploring/improving how the app is implemented in Räddningsmissionen’s current systems. The project will also explore how the companies are best greeted from a marketing perspective, by exploring the possibilities of developing a web-version of the app.

Räddningsmissionen is a non-profit organization that aim to redistribute society’s excess resources to the benefit of those in need. The products donated go to their store “MatRätt”, and soon to be opened “KlädRätt”. They are located in Rannebergen in the northeast part of the Gothenburg area.

1.2 Purpose

To analyze and improve the user experience for different users externally and internally inside the app, but also in the meeting of potential users from a marketing perspective.

1.3 Goal

To improve the user experience through user studies and iterative prototyping; with the end goal of increasing simplicity, user satisfaction, and the general number of users.

1.4 Research Questions

- What problems do the users encounter during the current process?
- What needs to be improved and what is missing with the current app?
- How do we get more users to the app?

1.5 Delimitations

Because Räddningsmissionen are based in Gothenburg, only potential customers in the Gothenburg area will be studied. The project will only analyze and develop the "ResursRätt" app, Räddningsmissionen other app "MatRätt" will not be included in this project.

2
Method

1. The project will start by identifying the app's different users: What groups use the app, who could use the app in the future, and who have access to read collected data. After different target groups have been identified a user study will begin.

2. The different users will be interviewed about their demands regarding the product as well as the different problems and areas of improvement revealed through usage. From these interviews list of problems, functions, and demands will be created. These will be used to create solutions and then evaluate said solutions. The users will also fill in the so called "User Experience Questionnaire" which will weigh different key-categories against each other to create an image of the experience.

3. Parallel to that internal studies and analyses will be done on the current app; how it is made and what is positive and negative about its current implementation.

4. After a pleasing amount of feedback has been collected the concept generation-phase will begin, where brainstorming will be used to create a large amount of ideas. These ideas will be based on the demands and functions that the users expressed in the user study.

5. The different concepts will first be evaluated with an elimination matrix which checks which of the functions are included in the concept and then decides which to keep and which to discard.

6. The kept concepts will then be used to create a morphological matrix which takes the sub solutions in the elimination matrix and lists them against each sub function. The new concepts will then be combined into different sub-solutions. A small number of sub-solutions will be created in such a they differentiate from each other as much as possible, in order to evaluate every possible solution fairly.

7. The new concepts can then be evaluated through a Pugh-matrix and a Kesselring-matrix as described in the theory. From these two methods the final concept can be chosen.

8. With the chosen final concept the prototyping can begin. During this stage prototypes will also be evaluated and tested with representatives from Räddningsmissionen. The prototypes will be created with the program Figma. Based on the thought and opinions from Räddningsmissionen, the prototype will be iteratively improved upon further.

9. Once satisfaction has been reached the final prototype will be tested by users, their opinions will be in mind when creating the final solution. During the process the list of demands and functions will be used to make sure that the solution is still solving the problem.

10. During the project the code will be analyzed, reevaluated and optimized in parallel to the user study.

3

Theory and Tools

This chapter provides an overview of the key technologies and theoretical foundations underlying both the design- and developmental phases of the project. It includes the methods utilized for analyzing user data, the tools and procedures used in creating prototypes, and the frameworks that supported the final implementation.

3.1 Miro

Miro is a software that works like a digital whiteboard [5]. Users can draw, write, and put post-it notes on a shared canvas. It is an ideal tool for organizing thought and ideas on an online platform. Miro was used in the project to create a mind map of the users' thoughts and suggestions. In this project each post-it note is a quote from a user.

3.2 The User Experience Questionnaire

The User Experience Questionnaire is a questionnaire designed to measure the attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty of a product [1]. The user weighs different keywords to achieve this.

3.3 Elimination Matrix

An elimination matrix is used to check if all functions are met in a concept and if the concept is worth keeping moving forward [3]. An example of an elimination matrix is shown in figure 3.1.

3.4 Morpheus: Morphological Matrix

A morphological matrix lists sub-functions and the different sub-solutions for each sub-function. It is then used to generate concepts from choosing a sub-solution for each sub-function. Morpheus is a software tool to create morphological matrices that was used in this report [2]. An example of a morphological matrix is shown in figure 3.2.

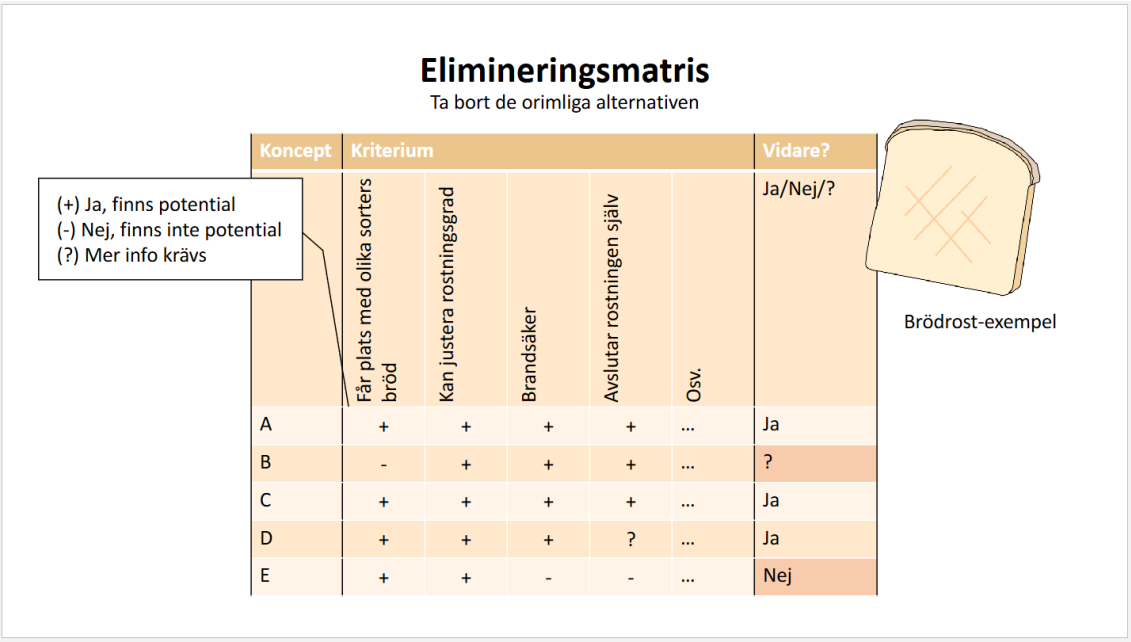


Figure 3.1: Example of an elimination matrix from Hulthén's lecture, page 69 [3]

3.5 Pugh-matrix

Pugh-matrix is a method used to evaluate concepts based on different chosen criteria [3]. One concept is chosen as the reference and the other are compared if it is worse or better on each criteria. When all criteria and concepts have been scored, the total score of each concept are summarized. The concept with the highest score is the best concept. An example of a Pugh-matrix is shown in figure 3.3.

3.6 Kesselring-matrix

Kesselring-matrix is a method used to evaluate concepts based on different weighted criteria [3]. Each concept is scored from 1-5 where 5 is the best on each criteria. The criteria are weighted so that their total weight is equal to 1. When all concepts are scored the score and the weight of each criteria is multiplied to create a result. The results are then added to create a total for each concept. The concept with the highest total is the best concept. An example of a Kesselring-matrix is shown in figure 3.4.

3.7 Figma

Figma is a collaborative UI/UX design tool used for digital interface design and prototyping [6]. The tool allows for interactive high-quality mock-ups without the need for complex software or extensive programming. Figma's output serves as references for translating visual designs into functional interfaces. The tool hosts support for multiple mobile-based layouts, assets, and presets which greatly reduce

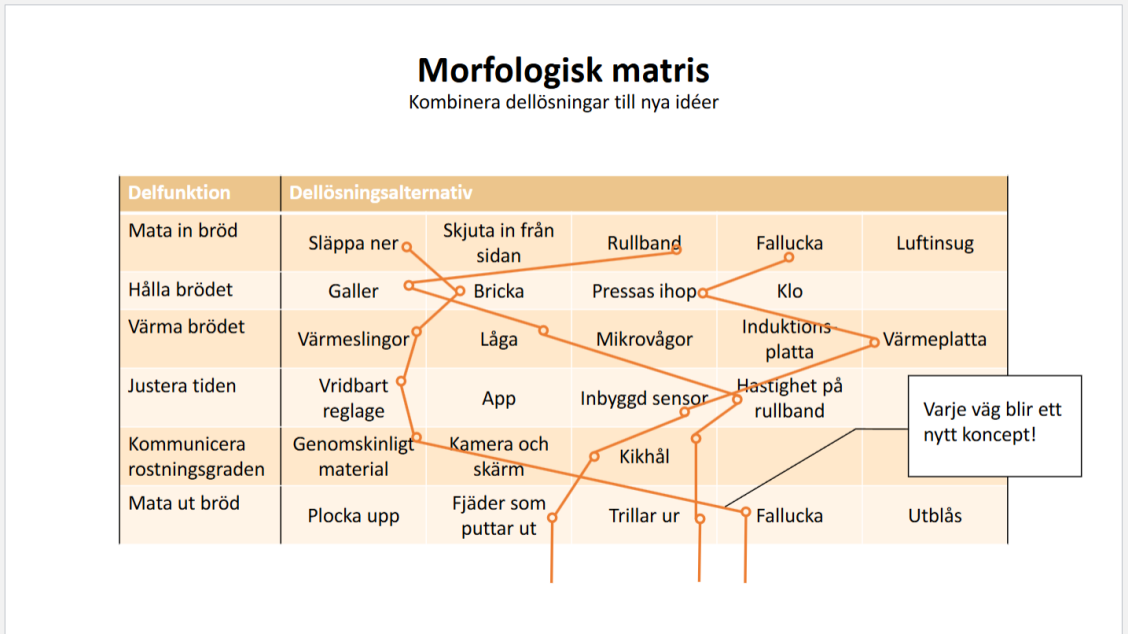


Figure 3.2: Example of a morphological matrix from Hulthén's lecture, page 62 [3]

the disparity between prototype and final product in app development.
Figma was used because it enabled the creation of dynamic prototypes that could simulate features suggested by users, eliminating the need for any prototype-coding until the final stages of the project.

3.8 Affinity Designer 2

Affinity Designer 2 (or as referenced in the report Affinity Designer) is an image editing software that can handle both pixel and vector graphics [7]. Affinity Designer was used to create various vector images in the Figma prototypes, for example button elements and custom icons.

3.9 .NET MAUI

.Net Multi-platform App UI (MAUI) is the cross-platform framework used in the app inhereted from Räddningsmissionen. It is designed for building native mobile and desktop application with a shared singular codebase [8]. MAUI ensured that the app's codebase (written in C# and XAML) was compatible with Android, iOS, macOS, Tizen and Windows without the need to rewrite the app for each platform. The framework has been generally available since May 2022, and includes many modern features such as hot-reload and the single-project structure which expedites coding and debugging. Its relative recency ensures constant patches and up-to-date frameworks, but also means that not every feature has a direct MAUI binding, and that performance may vary by platform.

Pugh konceptvalsmatris

Koncept för att visa tid och ställa alarm →					
Kundvärde	R	-	+	-	+
Uppfyllnad av huvudfunktionen	E	-	+	-	+
Uppfyllnad av delfunktioner	F	-	+	-	+
Robusthet	E	0	+	-	-
Andrahandsvärde	R	+	-	+	-
Materialkostnad	E	-	0	+	+
Komplexitet	N	+	-	+	-
Summa:	S	-2	+2	-1	+1

Figure 3.3: Example of a Pugh-matrix from Hulthén's lecture, page 66 [3]

3.10 C#

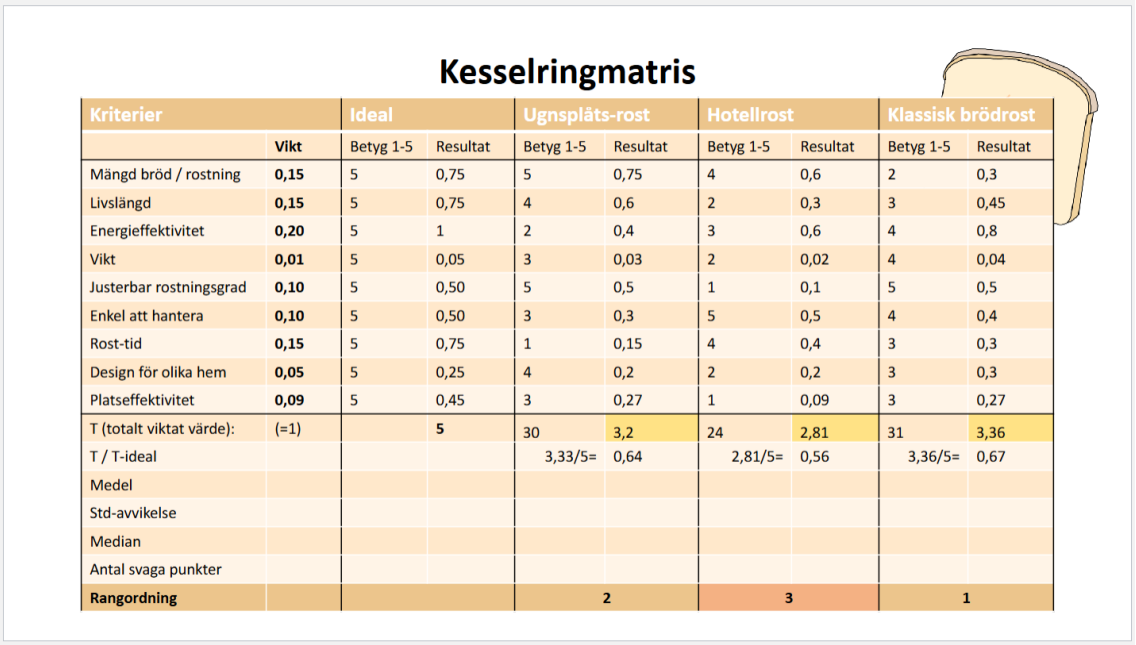
C# is an object-oriented language which serves as the primary backend language for .NET applications. As Microsoft, the producers of .NET MAUI, describe: "C# is a cross-platform general purpose language that makes developers productive while writing highly performant code"[9]. Using C# gives access to a rich array of tools (such as Visual Studio and IntelliSense) and thorough documentation.

3.11 XAML

Extensible Application Markup Language (XAML) is a programming language based on Extensible Markup Language (XML) used to define layouts and user interface elements (UI elements) [10]. XAML is standard for .NET applications and is responsible for the visual structure and elements of each page in the app.

3.12 Visual Studio 2022

Visual Studio 2022 is an integrated development environment (IDE) developed by Microsoft [11]. It has top-of-the-line support for .NET development, and tightly integrates C# features such as IntelliSense as well as XAML-hot-reload.



Kesselringmatris

Kriterier		Ideal		Ugnsplåts-rost		Hotellrost		Klassisk brödrost	
	Vikt	Betyg 1-5	Resultat	Betyg 1-5	Resultat	Betyg 1-5	Resultat	Betyg 1-5	Resultat
Mängd bröd / rostning	**0,15**	5	0,75	5	0,75	4	0,6	2	0,3
Livslängd	**0,15**	5	0,75	4	0,6	2	0,3	3	0,45
Energieffektivitet	**0,20**	5	1	2	0,4	3	0,6	4	0,8
Vikt	**0,01**	5	0,05	3	0,03	2	0,02	4	0,04
Justerbar rostningsgrad	**0,10**	5	0,50	5	0,5	1	0,1	5	0,5
Enkel att hantera	**0,10**	5	0,50	3	0,3	5	0,5	4	0,4
Rost-tid	**0,15**	5	0,75	1	0,15	4	0,4	3	0,3
Design för olika hem	**0,05**	5	0,25	4	0,2	2	0,2	3	0,3
Platseffektivitet	**0,09**	5	0,45	3	0,27	1	0,09	3	0,27
T (totalt viktat värde):	(=1)		**5**	30	3,2	24	2,81	31	3,36
T / T-ideal				3,33/5=	0,64	2,81/5=	0,56	3,36/5=	0,67
Medel									
Std-avvikelse									
Median									
Antal svaga punkter									
Rangordning				**2**		**3**		**1**	

Figure 3.4: Example of a Kesselring-matrix from Hulthén's lecture, page 69 [3]

3.13 Android Emulators

Android emulators are virtual devices that lets a host computer mimic the behavior of real Android hardware. Tools that allow developers to test and debug applications without needing the proper hardware are valuable in the app-making process. Software used was Android Studio followed by Visual Studio 2022's built-in Android Device Manager. Android Studio is Google's official IDE for Android development and a natural choice for Android development [12]; Visual Studio 2022's Android Device Manager however is integrated in Visual Studio (VS) and provides every feature required for app-development with better integration once a virtual device is created.

3.14 Model-View-ViewModel

Model-View-ViewModel (MVVM) is an architectural pattern used in many modern applications. It facilitates a clear separation between user interface and background logic which makes codebases more maintainable, modular, and easier to test.
The Model handles application data and logic, the View is responsible for the user interface (UI), and the ViewModel acts as a condiut and intermediary between View and Model. A core principle of MVVM is that the View should not communicate directly- nor be dependent on any specific Model-platform, which ensures that the app remains uncoupled and scalable. .NET MAUI has many built-in commands and mechanisms that support the MVVM pattern and streamline communication between the components; such as the ICommand-function for UI elements like buttons, and INotifyPropertyChangedto which automatically detect changes in element variables.

4

Process

4.1 Identifying target groups

Based on feedback from Räddningsmissionen the intended users of the app are company employees, in particular employed sustainability managers, store managers, or specialists in logistics. The targeted-groups are industries associated with food and/or clothing. Common examples include grocery stores, wholesalers, clothing stores and the like. When contacting different companies, one key aspect was that the users of the app could refer to a variety of people. The app therefore needs to be simple enough so that it is easy to use for anyone.

4.2 A semi-structured direct participating observation

To get a feel for the app and to find easily identifiable problems, a semi-structured direct participating observation was made [4]. This means that each step of the app was carefully observed, and all relevant thoughts and opinions where recorded. It identified and explored all areas of the app.

Figure 4.1: The first page of the original app

In figure 4.1, we see the initial page of the app. It has the logo that reassures the user that they opened the right app, and that the company is Räddningsmissionen: It explains the mission-statement of Räddningsmissionen and the reasoning behind it, and it reassures the user that the donation helps create a sustainable society. The presence of only a single button on the page, the donation button, helps guide the user's attention. It is clearly highlighted and guides the user to continue, and an arrow above it also strengthens this interaction. Lastly, the obligatory sponsor is displayed at the bottom of the page. All in all, the page is straightforward and the user does not have a problem moving forward in the app.

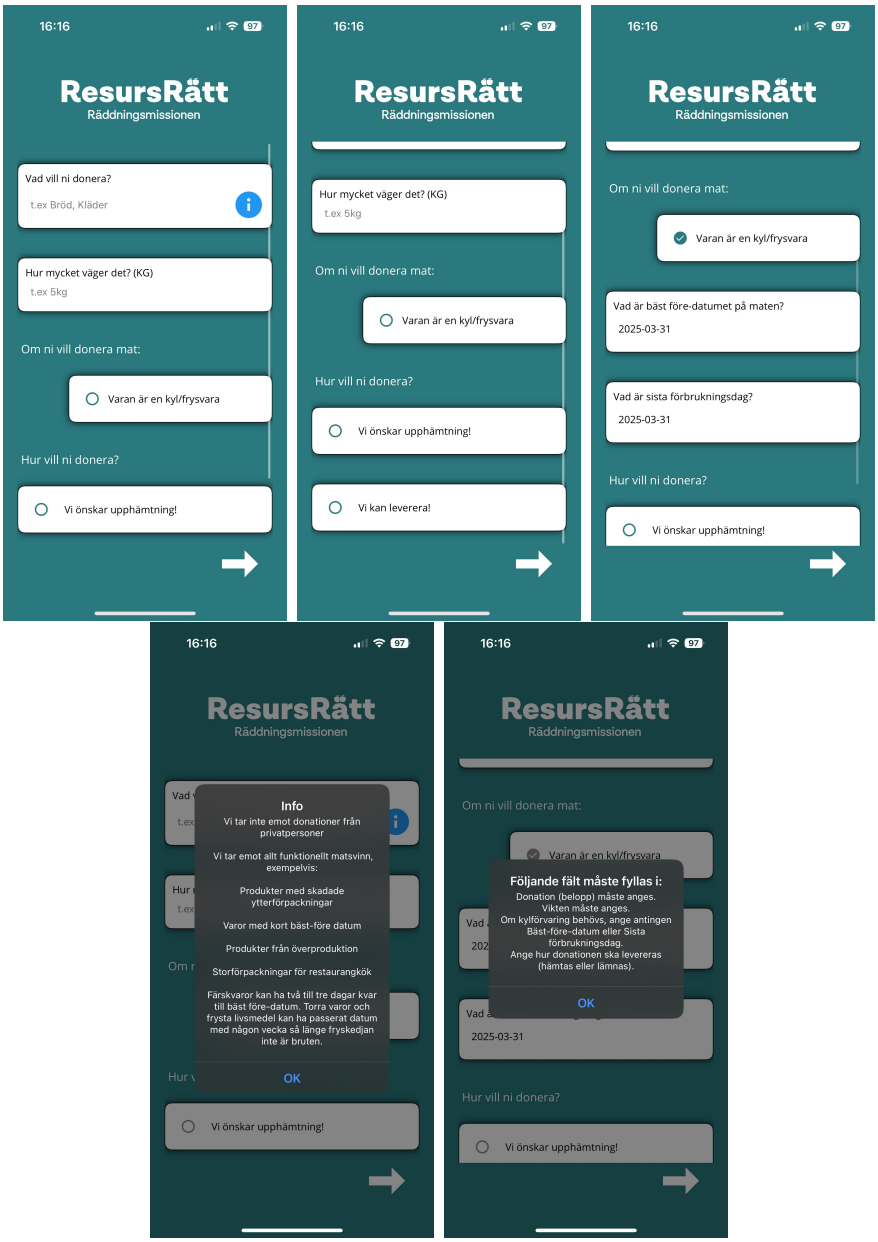


Figure 4.2: The donation page of the current app

In figure 4.2, we see the donation page and all the different pop-ups. The first question is "What do you want to donate?", which is a text box that the user must fill

out. To make the user understand what to write in the box, there is an information button that displays a more detailed explanation. However, the information box also first states that the app is not for individual people. It is counterproductive to have this fact be hidden away in an optional information box. It should be presented in a earlier stage; perhaps in the app store, promotion material or the first page of the app. The other information in the box is relevant and gives examples of what is usually donated. However, the examples are all food centered even though Räddningsmissionen accepts donations from a more varied assortment. The fact that items other than food can be donate is stated earlier however, where in the description they also mention that they accept clothes.

It might be slightly confusing in what way the company should fill in the boxes if they want to donate more than one type of item. For example food *and* clothes. The intended way is to fill out everything in the same string. In other words, one would write "Food and Clothes" or "Food, Clothes" in the box. The app then asks "How much does it weigh? (KG)", this could be rephrased to "How much does everything weigh in total? (KG)" to increase clarity. The wording of the question makes sure that the user does not write the individual weight of one of the items that they want to donate. Furthermore it clears up any confusion when donating multiple items in regards to if they should define the weight of every individual item. The app proceeds to ask if you want to donate food, and provides a check box that states "The product is a refrigerant or frozen goods". This seems to be in the wrong position or worded wrong. It would be unnecessary to ask this if the user specified something other than food. Another not is that all food does not fall in that category, as some food are dry or conserved, which can not be selected. If you do not check the box for refrigerant or frozen good it does not show the boxes for best-before date or expiration date. This would cause problems for products that have a best-before date, but do not need refrigeration. The last question the app asks is "How do you want to donate?" and gives the user a straightforward choice of pick-up by Räddningsmissionen or self-delivery.

If any box is not filled in as you complete the donation, you get an error pop-up which states what is missing. However text in the error message does not completely correspond to the boxes. A more visual indicator could be easier to understand by highlighting the missing boxes. To summarize this page could do with some redesigning and restructuring.

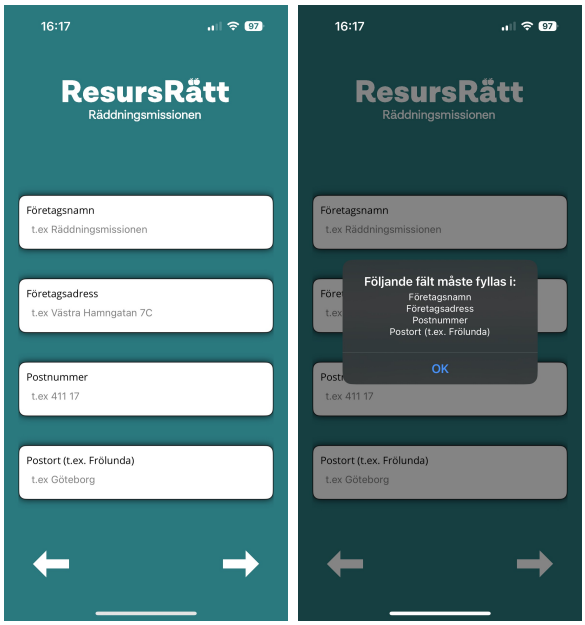


Figure 4.3: The company information page of the current app

In figure 4.3, we see the company page and the error pop-up. The company page asks the user information about the company for the purpose of delivery and registration. However, this does not account for companies that have different addresses for delivery and their office. A choice of either selecting same company address and delivery address, or otherwise writing down both would solve this issue. In the last question it also gives the example in the title as well as in the box. For continuity it would be easier to keep example-text inside the box. To increase clarity the page could start with stating that you are supposed to write your company information. The error pop-up corresponds to the boxes on this page unlike the previous, however, it would probably be more intuitive to visually show the error with highlights. All in all the page is rather straightforward with some minor adjustments needed.

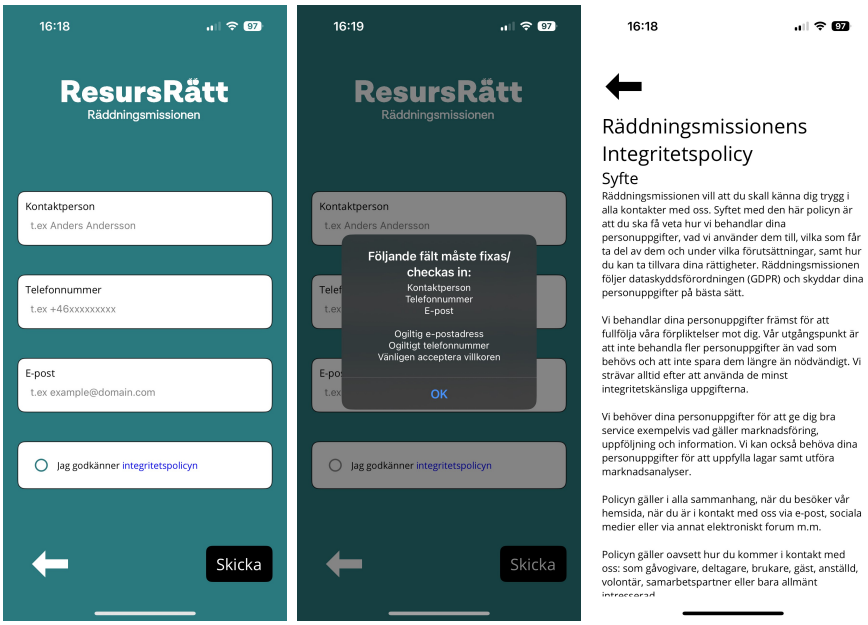


Figure 4.4: The contact page and integrity policy of the current app

In figure 4.3, we see the final page before sending the donation. This page asks who the user is, for the users contact information, and for them to agree with Räddningsmissionen's integrity policy. The error pop-up that appears when incorrectly filling out the form seems to be repeating itself, and is displayed in another order than the boxes. As stated with other error messages it would be more intuitive to highlight boxes with errors instead of a pop-up. The policy page is straightforward and you can read the policy easily.

Figure 4.5: The thank you page at the end of the app.

In figure 4.5, we see the final page of the app, the thank you page. It thanks the user and states relevant information of what will happen. The page then present

the user with two buttons. A button that says "Contact us" that opens an external browser and directs the user to their website. However, unlike what the button says it directs the user to the following link "`https://www.raddningsmissionen.se/donera-livsmedel`" where the user can donate to them. It would make more sense to link to "`https://www.raddningsmissionen.se/kontakt`" where you get contact information to Räddningsmissionen, since the user is told this is where one finds contact information. The other button called "Donate again" sends the user back to donation page.The page is straightforward with the minor issue of opening the wrong webpage.

The general appearance is very simplistic but also cheaply thrown together and poorly thought out. The buttons and boxes especially highlight this phenomena. A redesign of certain parts could reinforce the image of the app, and reinforce serious tone that Räddningsmissionen wants to portray. Some sort of progress tracker would also let the user know how many steps it takes to fill out, reassuring the user that it will not take a long time to fill out the donation.

4.3 Company feedback

To start, about twenty companies where contacted about providing input what they saw as requirements for the app. Only two of the twenty companies where willing to answer. The two companies will be anonymous in this report however it is worth mentioning that one company is from the food industry and the other is from the clothing industry. Both companies answered questions about necessary functions and information. The answers where then sorted in to different categories in the program Miro, see figure 4.6.

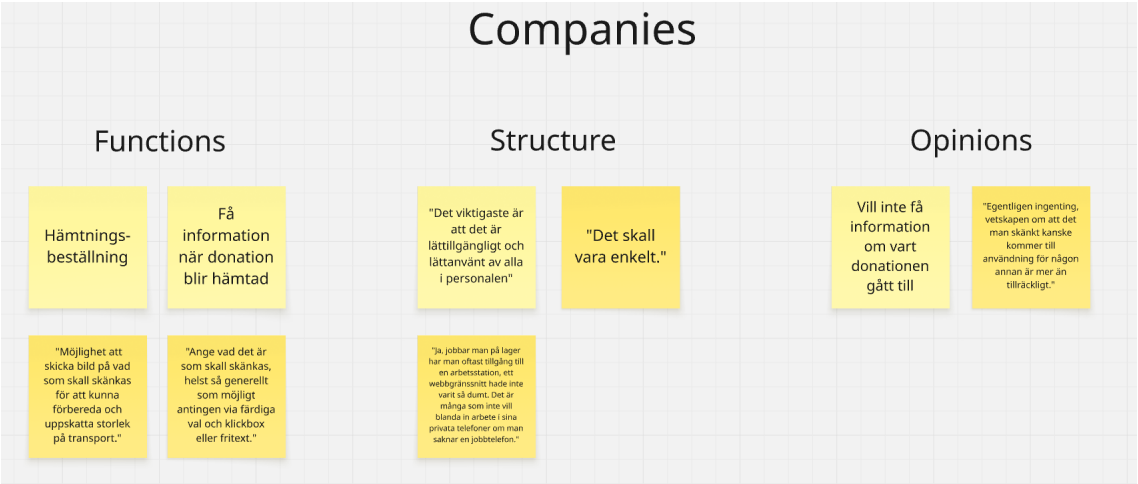


Figure 4.6: Miro of the company feedback, here the feedback are sorted in the three categories functions, structure and opinions.

In figure 4.6, the company responses are divided into three categories: Functions, structure, and opinions. The key functions that they needed were: pick-up delivery, information when a donation would be picked up, being able to send a photo of

what one donates, and to have guiding check-boxes to make it easy to fill in the donation. Under structure the most important aspect was for the app to be simple and easy to understand. Both companies expressed the opinion of not wanting to be informed where their donations went to.

4.4 User Tests

Seven users got to try the current app, where-after they got to express feedback and answer the following questions:

- "What is good currently?"
- "What is bad currently?"
- "Is something missing?"
- "Is something taking too much time?"
- "Is something unnecessary?"
- "Would you like it more if it instead was web-based? Why?"
- "How are the steps of the app? Do they work great?"

The answers where then categorized in Miro, in the following categories: Positive, Negative, Functions and Structure. Which are shown in figures 4.7-4.10.

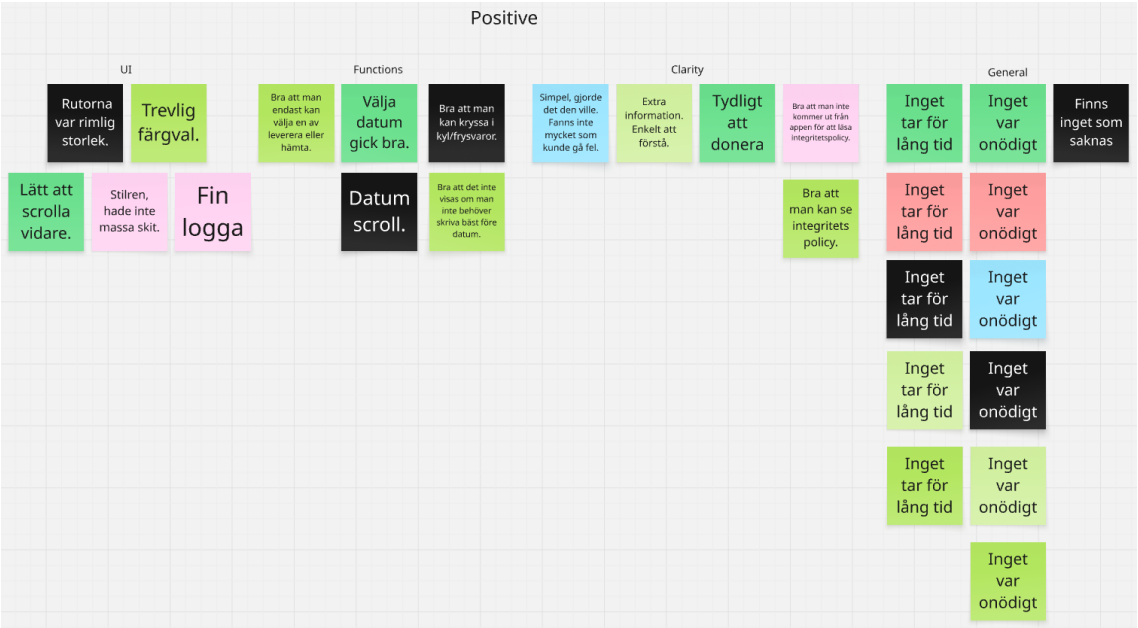


Figure 4.7: Miro of positive aspects of the current app based on user tests. The aspects are sorted into the following categories from left to right, UI, functions, clarity and general

In figure 4.7 the users positive thoughts about the current app are displayed. They expressed that no step took too much time and that no step was unnecessary. They liked that it was simple and easy to understand. In regards to individual functions, the users especially liked that the date selector was a date-scroll and not free text. Having to choose if the product needed a refrigerator or freezer was also a great feature, which meant that the user was not overwhelmed with many fields that

could otherwise be unnecessary. In regards to the user interface the boxes where of a great size and it was easy to scroll. The color choice and logo were stylish which was also positive for the users.

Figure 4.8: Miro of negative aspects of the current app based on user tests. The aspects are sorted into the following categories from left to right, uncertainties, annoyance, aesthetics/UI and miscellaneous.

In figure 4.8, the users negative thoughts about the current app are displayed. The thoughts can be categorized in the following categories: uncertainties, annoyance, aesthetics/UI and miscellaneous. The largest uncertainty the user faced was what to write in the weight-field. The app required an integer, but allowed you to write alphabetical letters in the space. The error-message only showed up after filling in everything else and attempting to move forward, which furthered confusion. The second largest uncertainty was that the user did not know or understand the difference between best-before date and expiration date. One user also felt uncertainty about the level of detail the user should provide in the box that asked what to donate. The biggest annoyance that the users experienced was that the keyboard did not go away unless they pressed "return". The keyboard did not disappear when they pressed outside the keyboard-area, which users expected. There was also a lack of a button that says "done". Another annoying factor was that the link in the integrity policy was not clickable, so the user had to copy and paste it in to a browser. In regards to the aesthetics and UI, the users pointed out a bunch of different issues; such as unclear rubrics, spacing between elements, scroll-bars, and shadows. Other thing the users commented on was that the app saves data between uses without explicit permission, and that it takes a lot of time to donate if one wants to give away mulitple types of items.

In figure 4.9, the user-suggested functions are displayed. Most of them are regarding the donation input, but there are also suggested functions about UI/backend, and order information. The two most sought-after functions are to provide additional information, and to specify the quantity of the donation in another way than weight. Being able to categorize donations, or to scan bar-codes are also wanted features. Functions regarding UI/backend are for example auto-filling city based on postal codes, consequent errors that pop up right away when a field is wrong, and a button to clear and restart the donation process. In regards to order information, many users said that some sort of feedback when sending the order would be appreciated. Either an email saying that the order is being processed, or getting a rough estimate of when the order will be picked up.

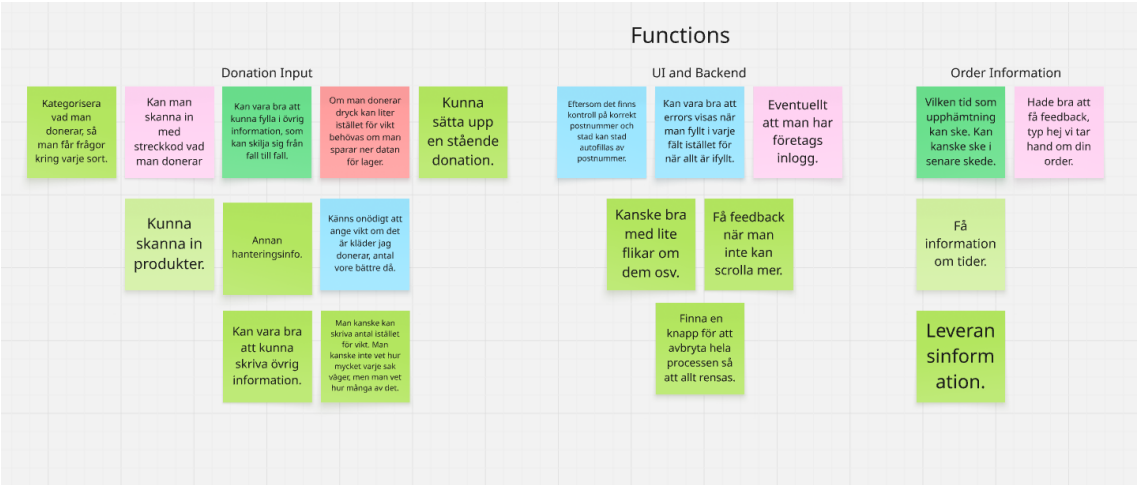


Figure 4.9: Miro of functions suggested in user tests. The functions are sorted in to the following categories from left to right, donation input, UI and Backend, and finally order information.

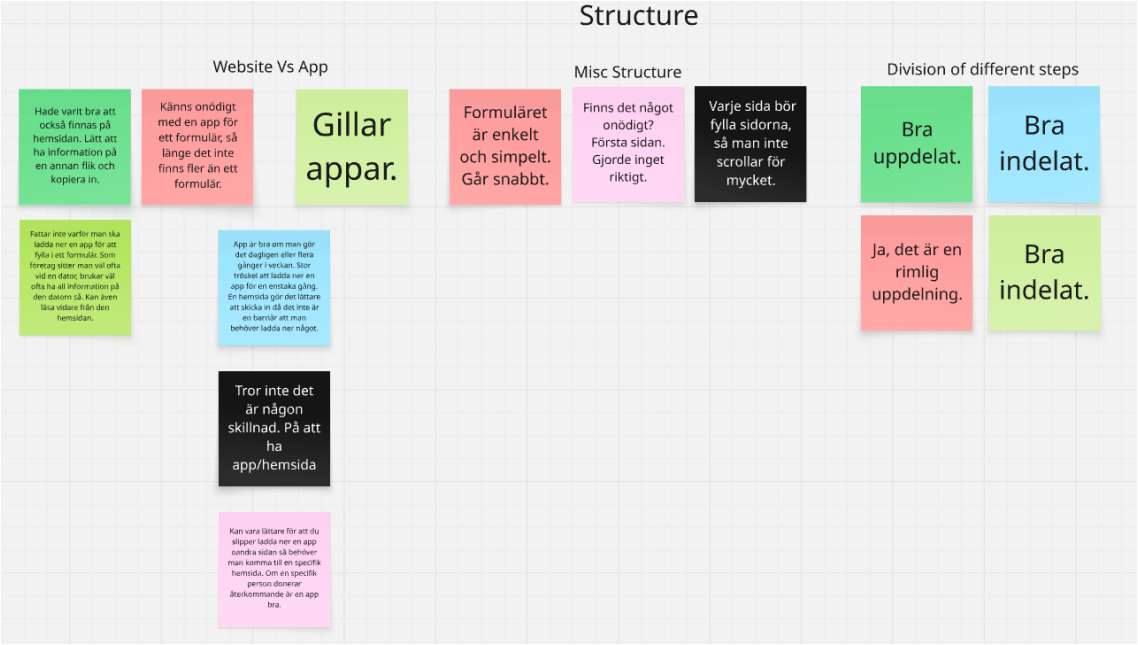


Figure 4.10: Miro of opinions from the user tests about structure. The opinions are categorized as follows from left to right, website vs app, miscellaneous structure and division of the different steps.

In figure 4.10, the users reflected on the structure and their opinions about having an app versus having a website. First and foremost the users stated that the donation-steps where in a good order and contained. The structure facilitated speed and simplicity which was also appreciated. One user said that the starting page was unnecessary, and another said that they wanted pages that had minimum scrolling. The opinions about having an app or a website where split. Some liked apps, while other preferred websites. An individual that liked the website-idea more said: "...Easy to have all the information on another window and just paste into the form." and another stated: "...As a company you usually sit at a computer and often have the information needed on that computer.". Some people saw opportunities with having both a website and an app. An app was good for frequent donations meanwhile a website was good for one-off donations.

The user experience questionnaire, as explained in chapter 3.2, was used to compile user data. This created an insight into what the user thought of their experience and could later be used to compare to the new version of the app. In figure 4.11, the mean result based on all the users questionnaires are displayed.

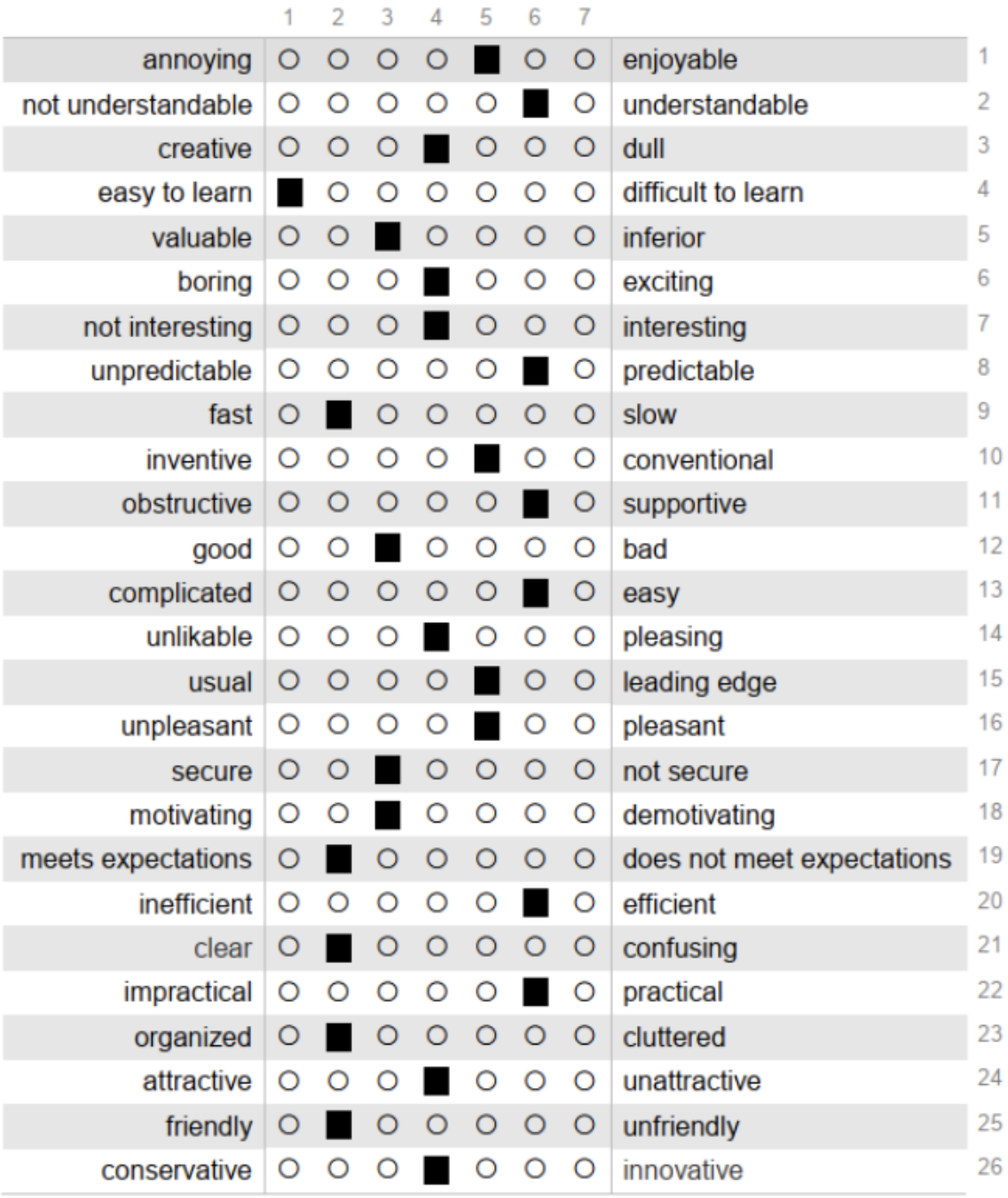


Figure 4.11: Mean Result of the User Experience Questionnaire

4.5 Lists of problems, functions and demands

From the user tests the users pointed out a few problems which are shown in table 4.1. Each problem are weighed with a priority. The priority is from 1-5 where 5 is the most important and 1 is the least important. The priority degree is determined from how many users stumbled on the problem and how much of a obstacle it is for the user. Aesthetic problems are less prioritized since it does not stop the user from donating.

Problem	Priority
General	
The keyboard does not disappear as expected	5
Uncertainty what to write in "Weight"	5
Unclickable link in Integrity policy	4
Uncertainty between best-before date and expiration date	4
Could miss that it is scroll	1
Unclear where one should deliver it to	2
Some check boxes that are related are to far apart	2
The date field get greyed out when you fill the other which could indicate that you only fill out one	3
Uncertainty of what should be inputted in the donation box	4
Uncertainty about what Räddningsmissionen do	1
Unclear text on the first page	2
Only states that the app is for companies in a hidden away information window	2
Food-centered even though they accept other products	4
Error does not show after one has filled in everything on a page	3
Unclear error messages	3
Not possible to fill in a different company adress and delivery adress	3
Incorrect link to contact information	2
Aesthetics	
The scrollbar is overlayed over the boxes	1
The refrigerator / freezer box is not centered	2
Ugly boxes	1
No clear headlines	2
Same font size of the questions and answers	2
Small x for the numbers in the telefon prompt, instead of large X	2
Clipping shadows on boxes	2
Typography error in information pop up	2

Table 4.1: Table displaying identified problems.

In table 4.2, the functions of the app are listed. These functions are based on the current app, the requirements from Räddningsmissionen, and feedback from users. Each function is either the main function, a sub-function or a support function. Every function also has a priority based on how important they are and if they are realistically achievable.

Function	Boundary	Class	Priority	Comment
Donate things	Only from companies	MF	Necessary	
Categorize donations		SUPPF	Desired	Food, Clothes, Other
Scan products	Barcode	SUBF	Worth Considering	
Provide additional information		SUPPF	Desired	
Set prefix	Metric	SUBF	Worth Considering	To define how much one donates
Create standing donations		SUBF	Worth Considering	
Autofill City/Postalcode		SUPPF	Worth Considering	If you fill in city it fill autofill postalcode and vise-versa
Display errors		SUPPF	Necessary	
Login Service		SUBF	Unnecessary	Could be implemented in a more complex system
Additional pages		SUBF	Worth Considering	About page, donation history, etc.
Reassure user		SUBF	Desired	That the user has seen everything and understood the assignment
Provide order feedback		SUPPF	Desired	Order and delivery information
Provide company information		SUPPF	Necessary	
Provide contact information		SUPPF	Necessary	Both from customer and from Räddningsmissionen
Provide visual	jpeg, png	SUBF	Worth considering	Photo of the donated product
MF = Main Function, SUBF = Sub-Function, SUPPF = Support Function				

Table 4.2: Table of functions

In table 4.3, different demands regarding delivery, information, process, platforms and semantics are displayed. These are both demands from Räddningsmissionen, and the users. They are also prioritized from 1-5; where 5 is the most important and 1 is the least important. Here key functionalities are prioritized, such as the app working on different platforms and having delivery options.

Demand	Specification	Priority
Delivery	Choice for pick-up	5
	Choice for self-delivery	5
Information	Pick-up information	4
	Self-delivery information	4
Process	Simple	4
	Accessible	4
	Fast	3
Platforms	Continuity across platforms	5
	Android	5
	iOS	5
Semantics	Express Professionalism	3
	Express Simplicity	3

Table 4.3: Table of Demands

These lists will be used to create concepts in the next step but also for evaluating said concepts and the final product.

4.6 Generating concepts

To choose and create concepts for prototyping the following steps have been taken: Firstly to brainstorm based on the problems, functions and demands defined in tables 4.1-4.3. This creates a concept-catalog that can be evaluated with an elimination matrix to see what functions are already incorporated, and which concepts should be implemented in the morphological matrix. The morphological matrix then takes the sub-solutions of each kept concept and lists them against each sub-function. Then new concepts are picked by choosing one sub-solution each in the list. The purpose is to get new concepts that incorporate design ideas from different base ideas. The new concepts must importantly differentiate from each other as much as possible to see interesting results when later evaluating them with a Pugh-matrix and Kesselring-matrix. The Pugh-matrix and Kesselring-matrix both use different criteria that seem relevant for the product, and then evaluates each of the concepts based on that. In the end one or two concepts will be the winners and their positive attributes can be incorporated in a prototype.

4.6.1 Brainstorming

The following concept catalog was made through brainstorming based on the problems, functions and demands defined earlier.
Concept A, as shown in figure 4.12 shows the app step by step. The first step is fairly similar to the original but points out to rewrite the text to be more informational. The main point of the concept comes next which is to categorize the donation. It has three categorizes "Food", "Clothes" and "Other", based of what you selected it drops down questions under said category. Choosing one or more categorizes is possible

and they will follow after each other. If food is selected it asks what food and what is the total weight, the user fills the two boxes. It then asks if the food requires a refrigerator, freezer or if it needs neither. It then asks the user what the best-before and expiration dates are. The user is provided a information button that explains the difference between them. If Clothes or Other is chosen the app asks the user to provide the item's name. The user can then choose quantity or weight to measure the quantity of the item(s). Lastly the user chooses between pick-up delivery or self-delivery. An arrow then sends the user to the next page. This page has a rubric that says "Company Information", it asks the user the company's name and also provides the user with a choice of same delivery address as company address. If yes the app will make the user fill in two addresses labeled respectively. When postal code is inputted the app automatically fills in the city. The next page has a rubric called "Contact Information" and asks the user for their name, telephone number and e-mail. Finally it asks the user to agree to the integrity policy. If the user clicks on the integrity policy a popup appears that the user can read. Finally a large button that says "SEND" is at the bottom that sends the order and brings the user to the thank you page. The last page is visually identical to the original "thank you" page. The app also sends an e-mail to the user's provided email-address that contains the submitted order information. This is to reassure the user that their request has gone through the system correctly. If anything is inputted in a field that would create a error, the app will mark the field red and write an error message under it.

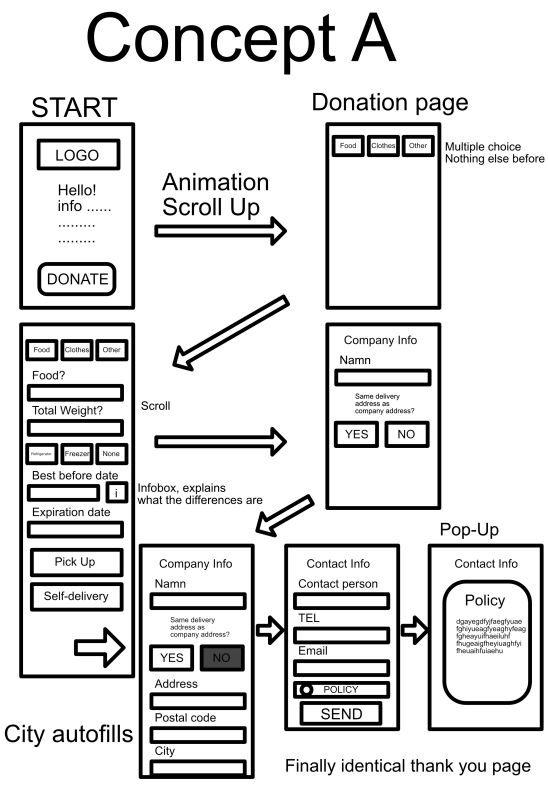


Figure 4.12: Concept A, full step by step guide using different categories to help the user along and ask relevant questions.

Concept B, as shown in figure 4.13 shows the app step by step. It starts with a similar first page as the original app. It then presents the main concept: The user is presented with a "plus"-button which when pressed adds one item. The user then writes the item name and fills in the weight. The user is then presented the choice of needing refrigeration which in this case the user ignores. The user can then choose between adding another item with the "plus"-button or go forward with the arrow. If choosing the plus, the user gets another element that is the same as the previous one. If the user chooses that the item needs to be refrigerated it then they get to fill in best-before- and expiration-date of the item before continuing. The user can add another item, or go forward with the arrow once more. Moving forward, the user gets to choose pick-up delivery or self-delivery. Following that, the user fills in information about the company and contact information. Finally the user sends the information through a final send button and the app thanks the user.

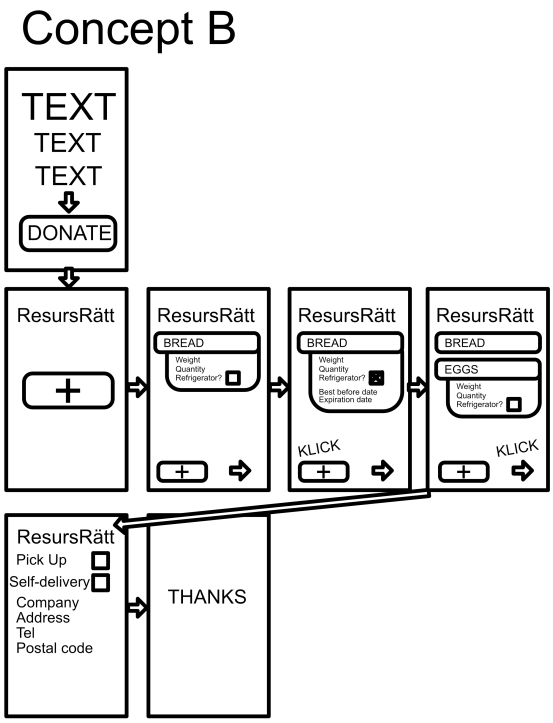


Figure 4.13: Concept B, step by step guide with the main feature being able to add multiple items.

Concept C, as shown in figure 4.14 shows how to fill in the donation part of the app. The rest of the app is a mixture of Concept- A and B. Unlike the current app the user is presented with predetermined choices to easily fill out correctly. If the wanted choice is not on the list it then the app lets the user fill in the field themselves.

Concept D, as shown in figure 4.14 shows how to fill in the donation part of the app.

The rest of the app is also a mixture of Concept- A and B. The user is presented with three categories: food, clothes and other, which when clicked drop down a list of subcategories, for example food that need refrigerators Under the food category. The user then fills in what the item is and other necessary information like in concept A or B. Multiple categories and subcategories can be chosen.

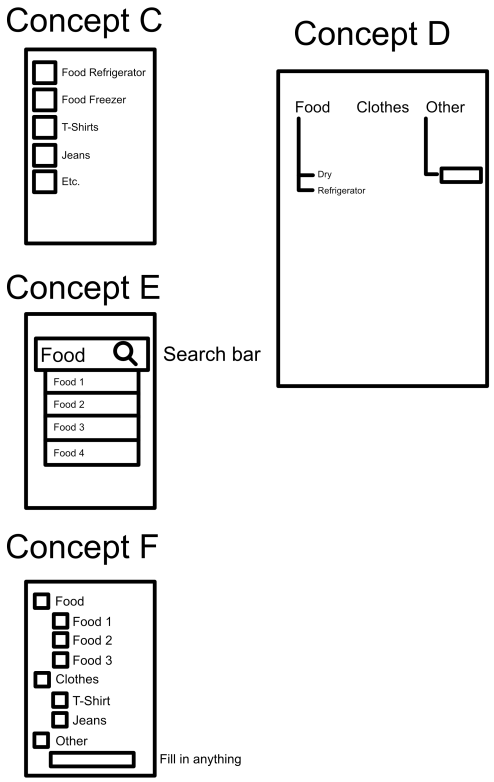


Figure 4.14: Concepts C, D, E, and F. C is in the top-left which has predetermined items to choose from. D is in top-right which shows three categories that open up to a tree liked layout. E located to the middle-left, shows a search bar with suggestions on what is typed. F is located bottom-left and contain check-boxes that in different categories.

Concept E, as shown in figure 4.14 shows how to fill in the donation part of the app. The rest of the app would copy either Concept A or B. The user is presented with a search bar that has examples that can be auto filled which helps the user provided needed information. If the search bar does not find the item then it then creates a custom field.

Concept F, as shown in figure 4.14 shows how to fill in the donation part of the app. The rest of the app is either like concept A or B. This concept is built around check-boxes: There is the main three categories food, clothes and other. In each category there are predetermined options that the user can choose, if none of them match the item the user can fill in a field that is empty.

Concept G, is of a new function that is not present in the current solution or other concepts. It is an menu button in the upper corner where the user gets a menu that slides in from the side to show the following options: Donate, Donation History, Contact Us, About Us and Start Over. Donate gets you back to the form where you where. Donation History shows all order the user has sent through the app. Contact Us provides a page with relevant contact information to Räddningsmissionen. About Us provides a page that explains what Räddningsmissionen does. Start Over clears the forms including donations, contact information and company information to get a clean slate to start the donation process over.

Concept H, as shown in figure 4.15 show all the different pages of the app. On the first page of the app concept H has added a additional button compared to the original that provides a pop up with more detailed information about Räddningsmissionen. By pressing "Donate" the user is presented with the page "Donation Information" where the user is given 3 check-boxes and an additional field where the user can provide additional information about the items. This field is optional and is available for the user to provide vital information the app otherwise would miss. If one or more boxes is checked it ask the user under each category what the item is and what it weighs. If food is checked it also asks if the item needs a refrigerator, freezer or none through three check-boxes. Lastly it asks for best-before and expiration date, it also provide a information button to clarify what each means. The user is then sent to the page "Delivery page": Here the user user is presented with a choice of pick-up delivery or self-delivery. If pick up delivery is chosen it then asks for the address and auto-fills the city based on the postal code. The next page is "Company Information": Here the user is asked what company it is and who the contact person at the company is. Finally the integrity policy is shown, which can be read through a pop-up. After pressing send, the app brings you to the "thank you"-page similar to the original app, and also sends an e-mail to the provided e-mail address with order information. The app errors are highlighted as the user progresses and explain what is wrong under each field. There is a visible scroll-bar to the right on each page and at the bottom there are three dots designed to show how much progress the user has done.

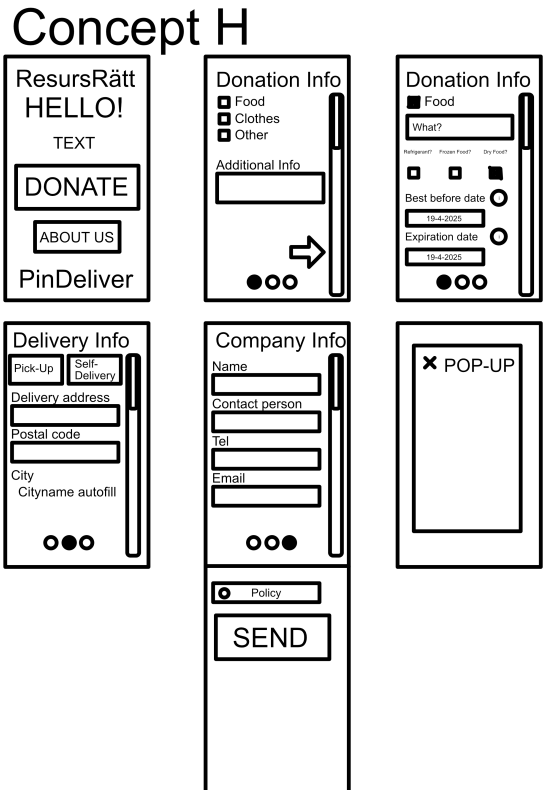


Figure 4.15: Concept H, has three different checkboxes and each ask questions depending on what is chosen and also provides the option to input additional information and an overlay that pops up when pressing an information button.

4.6.2 Elimination Matrix

To evaluate each concept from the brainstorming, an elimination matrix was used to determine what functions where met from the list of functions in table 4.2. A plus means that it meets the function and is kept, where a minus does not meet the function. Some of the concepts where more flawed then others, however every concept was kept going forward to the morphological matrix since they showcased unique solutions that could show great result in the morphological matrix.

Concept															Decision
	Donate Things	Categorize donations	Scan Products	Provide additional information	Set prefix	Create standing donations	Auto-fill City/Postal-code	Display Errors	Additional pages	Reassure user	Provide order feedback	Provide company information	Provide contact information	Provide visual	
A	+	+	-	-	+	-	+	+	-	+	+	+	+	-	+
B	+	-	-	-	-	-	+	+	-	+	+	+	+	-	+
C	+	+	-	-	+	-	+	+	-	+	+	+	+	-	+
D	+	+	-	-	+	-	+	+	-	+	+	+	+	-	+
E	+	+	-	-	+	-	+	+	-	+	+	+	+	-	+
F	+	+	-	-	+	-	+	+	-	+	+	+	+	-	+
G	+	+	-	-	+	-	+	+	+	+	+	+	+	-	+
H	+	+	-	+	-	-	+	+	+	+	+	+	+	-	+

Table 4.4: Elimination Matrix of the concepts from the concept catalog

4.6.3 Morphological Matrix

The morphological matrix takes each function that was either necessary or worth considering from the list in 4.2. Then each sub-solution, for each sub-function, from each concept is listed as shown in table 4.5.

The result of the morphological matrix are the concepts 1-5 which are shown in table 4.6. These where chosen because they had the most promising features and included key differences how the donation process was made. This made them more interesting to compare.

Sub-functions	Sub-solutions				
Categorize Things	Three categories then questions depending on category	List of predetermined items	Tree drop-down	Search bar	Check-boxes with sub check-boxes
Provide additional information	Blank field				
Set Prefix	Depending on category	Choice			
Auto-fill Postal Code/city	Auto-fills city name depending on postal code				
Display Errors	Highlighted errors with descriptive text underneath				
Additional Pages	Menu button	Read more buttons			
Provide order feedback	Send an e-mail	Donation History			
Provide company information	Fields, choice between delivery address and company address	Fields			
Provide contact information	Fields				
Misc	List items. Plus button to add to the list.				

Table 4.5: Morphological Matrix

Sub-functions	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
Categorize things	Check-boxes with sub check-boxes	Search box	Tree drop-down	List of pre-determined items	Three categories then questions depending on category
Provide additional information	Blank field	Blank field	Blank field	Blank field	Blank field
Set prefix	Choice	Choice	Choice	Depending on category	Choice
Auto-fill Postal code/City	Auto-fills city name depending on postal code	Auto-fills city name depending on postal code	Auto-fills city name depending on postal code	Auto-fills city name depending on postal code	Auto-fills city name depending on postal code
Display Errors	Continuously highlighted errors with descriptive text underneath	Continuously highlighted errors with descriptive text underneath	Continuously highlighted errors with descriptive text underneath	Continuously highlighted errors with descriptive text underneath	Continuously highlighted errors with descriptive text underneath
Additional Pages	Read more buttons	Menu button	Menu button	Read more button	Menu button
Provide order feedback	Donation History	Donation History	Send an e-mail	Donation History	Send an e-mail
Provide company information	Fields	Fields, choice between delivery address and company address	Fields, choice between delivery address and company address	Fields	Fields, choice between delivery address and company address
Provide contact information	Fields	Fields	Fields	Fields	Fields
Misc	List items. Plus button to add to the list.	List items. Plus button to add to the list.	List items. Plus button to add to the list.	List items. Plus button to add to the list.	List items. Plus button to add to the list.

Table 4.6: Result of the Morphological Matrix

4.6.4 Pugh-Matrix and Kesselring Matrix

The Pugh-matrix is shown in table 4.7, and shows each concept compared to concept 1. The criteria are chosen based on how important they where for the project. The result of the Pugh-matrix shows that concept 1 followed by concept 5 are the most promising concepts.

Criteria	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
Customer Worth	R	0	-	-	-
Meeting Main Function	E	0	0	0	0
Meeting Sub Functions	F	-	0	-	0
Complexity	E	0	-	+	0
Efficiency	R	+	0	-	0
Intuitive	E	-	-	0	0
Continuity	N	+	-	0	0
Clarity	C	0	-	-	0
Integration	E	-	-	-	0
Net Worth	0	-2	-6	-4	-1
Ranking	1	3	5	4	2

Table 4.7: Pugh-Matrix

In table 4.8 the kesselring-matrix is shown. It evaluates each concept based on the same criteria as in the Pugh-matrix. The Kesselring-matrix however weighs each criteria where in this case the criteria meeting main function, meeting sub function and customer worth are most important. R stand for result in the figure and is calculated by multiplying the weight of the criteria by the score from 1-5. In the end all resulting scores are added together to create the total, T. Then they idea with the highest total gets rank 1, and the second to highest get rank 2 and so on. The best concepts according to the Kesselring-matrix are concept 1 followed by concept 2.

Criteria		Ideal		1		2		3		4		5	
	W	1-5	R	1-5	R	1-5	R	1-5	R	1-5	R	1-5	R
CW	0.15	5	0.75	5	0.75	5	0.75	3	0.45	2	0.3	4	0.6
MMF	0.2	5	1	5	1	5	1	5	1	5	1	5	1
MSF	0.15	5	0.75	4	0.6	3	0.45	4	0.6	3	0.45	4	0.6
Com	0.08	5	0.4	4	0.32	4	0.32	3	0.24	5	0.4	4	0.32
E	0.1	5	0.5	4	0.4	5	0.5	4	0.4	2	0.2	4	0.4
I	0.1	5	0.5	4	0.4	3	0.3	3	0.3	4	0.4	4	0.4
Con	0.07	5	0.35	3	0.21	4	0.28	2	0.14	3	0.21	3	0.21
Cla	0.08	5	0.4	4	0.32	4	0.32	3	0.28	3	0.28	4	0.32
Int	0.07	5	0.35	4	0.28	1	0.07	3	0.21	2	0.14	4	0.28
T	1	45	5	37	4.28	34	3.99	30	3.62	29	3.38	36	4.13
Rank				1		3		4		5		2	

W = Weight, R = Result,
CW = Customer Worth, MMF = Meeting Main Function,
MSF = Meeting Sub Function, Com = Complexity, E = Efficiency, I = Intuitive,
Con = Continuity, Cla = Clarity, Int = Integration, T = Total

Table 4.8: Kesselring Matrix

The concept generation results in the creation of a prototype that mixes concept 1 and 5. It will have three main check-boxes that show different sub-check-boxes that then asks different questions about the donation based on what was selected. An additional information-field will also be present, and the choice of prefix will depend on category. The food category will have weight while the other categories will have a drop-down menu to select desired prefix. The final concept will have an auto-fill function for the city based on the postal code. The app will also continuously highlight errors in red and provide a descriptive text underneath the field with an error. A menu button will be implemented and additional pages with information will be provided at key areas such as best-before-date and integrity policy. The app will also have a feature in the menu button that lets the user see the history of donations made through the app, and when a user makes a donation they will get an e-mail that reassures the user that their order went through. The final concept will ask between pick-up delivery or self delivery and will only ask for address if the pick-up delivery is chosen. One other key feature is the ability to add more then one product in a single donation. The user will see a list of the items they want to donate.

4.7 Prototyping

The first prototype was made based on what was decided from the concept generation. The first prototype prioritized functionality over aesthetics. The prototypes where made in Figma to easily change the layout during production. The first prototype was showcased to Räddningsmissionen: It being simple and malleable was key to indicate that the prototype could be changed based on their feedback. They were

overall pleased and suggested some changes that was implemented in the second prototype.

4.7.1 Prototype v1.1

The first prototype v1.1 focused on functionality and the overall layout, meaning where the buttons should be- and what size they would take. In figure 4.16 the starting page along with the menu and donation history page are displayed. The starting page is changed minimally from the original, the two key differences are the menu button that will be displayed on all pages and the donate button which has been made larger. The menu is a new feature that was not in the original app. In the menu the user can change language, view their donation history which is displayed in a list with dates and what was ordered. The menu also features a "Start over" button that takes the user back to the start page from any page of the app and clears all inputs that was previously filled in.

Figure 4.16: v1.1 of the start page, menu and donation history

In figure 4.17 the donation steps are displayed. This part has been changed the most from the original app. The two key differences are that each item is categorized and asked only relevant questions based on said category. The other key difference is that the user now can add multiple items to the donation that do not need to have the same additional information as each other. For example, the user can add bread with a certain amount and best before date, and also at the same time add milk; which requires refrigeration with another amount and best before date. Firstly the user is asked what category the item is, either "Food", "Clothes" or "Other". Then the user gets questions based on what category was chosen. If the food category is chosen the user gets to choose between "Refrigerant", "Frozen Foods" and "Dry Foods". The user then fills in a text field to answer what it is and the field gives examples such as "Minced Meat, Butter, Milk" depending on the sub-category. Then the user is asked how much the total amount of items weighs in kilograms. The user then gets to fill in best-before-date and expiration-date. If the user does not know

the difference the user can click on the "i" button to get a pop-up with information about the differences between the two, and what to input in each field. When the user is done reading the user can press the X-button or click outside the pop-up. The user can then input the optional additional information field. Finally the user chooses between adding another item to the donation with the plus button or go forward with the forward arrow. If the user instead chose the clothes category the user is asked what they donate for example "T-Shirt, Jeans, Shoes", the amount of said clothing and optional additional information. If the user chose the other-category they are asked what it is followed by the quantity, where the user can choose their own prefix: For example kilograms, quantity or volume. The user can also fill in the optional additional information. If the user adds an item a list will be created at the top where the user can view all inputted items. Here the user can remove an item or edit them before sending in the final donation. Unlike the current app the user does not choose the delivery method in this step. They instead move on to the next step.

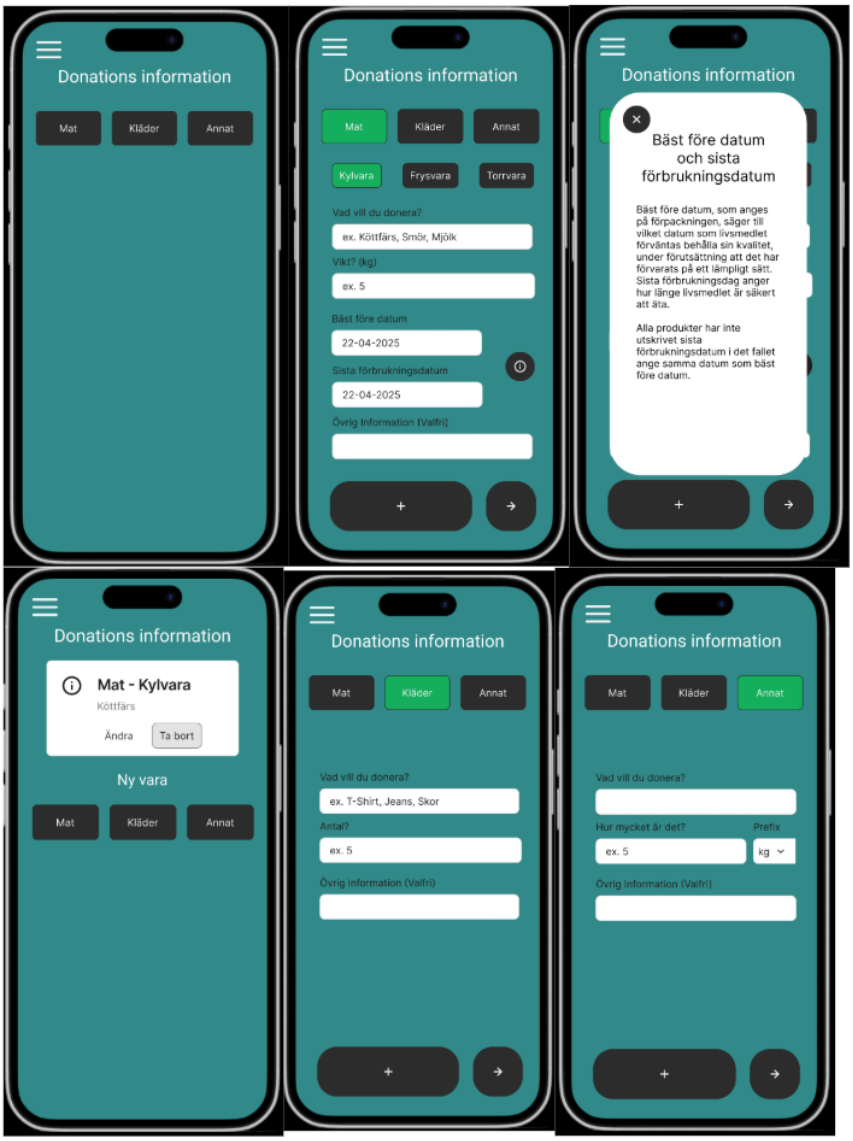


Figure 4.17: V 1.0 of the donation page

The next step of the app is to fill delivery information, which is shown in figure 4.18. The user is asked to choose between pick-up delivery or self-delivery. If self-delivery is chosen they only need to fill in their company name. If pick-up delivery is chosen they also need to fill in their company name, but additionally the delivery address and postal code. When the postal code is inputted, the app automatically fills in the city in the city box. The user can always move back to edit the donation but if everything is filled in the user can choose to move forward in the app, to the final step.

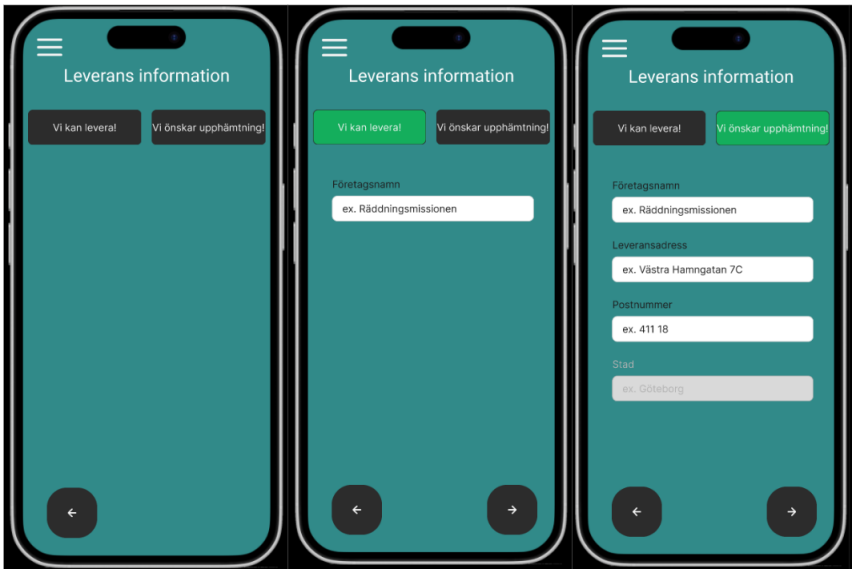


Figure 4.18: v1.1 of the delivery information page

The final step of the app is to fill in the contact information so that Räddningsmissionen can reach the user. In figure 4.19 the page is shown. The user inputs their name, telephone number and email. Before the user can send the order they need to agree to the integrity policy of Räddningsmissionen. By pressing the checked box they agree to policy an can send the order. If the user want to read through the policy they can press the "i"-button and get a pop-up which displays the policy. The user can scroll through the policy to read everything, and when the user is done they can either press the X-button or click somewhere outside of the pop-up. The user has a final chance to go back and edit something before clicking the big "SEND"-button.

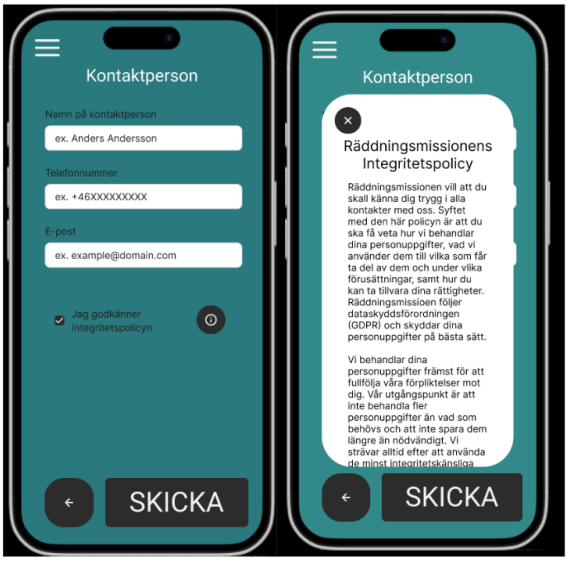


Figure 4.19: v1.1 of the contact information page

The final page, shown in figure 4.20 is similar to the original app but has slightly

larger buttons and the "Contact us" button leads to Räddningsmissionen's contact page on their website instead of their donation page.

Figure 4.20: v1.1 of the 'thank you' page

As previously mentioned this prototype was shown to Räddningsmissionen and they provided the following feedback: The prototype is very well thought out and the changes are well motivated. Räddningsmissionen suggested that the idea to ask for amount or being able to fill in their own prefix was very interesting, however they said that they could count when they get the items and that it is more important for them to know the approximate volume so they suggested that the user should be asked for clothes and other items to be described in volume. An example of said description could be 'X boxes' with described measurements or 'X pallets'. They also wanted the "Contact us" button to be a mail-to link instead of going to their website and also suggested that there should be a "Contact us" button in the menu as well.

4.7.2 Prototype v1.2

The Prototype v1.2 is the more refined version of v1.1 with changes based on the feedback recieved at the meeting with Räddningsmissionen. Prototype V.1.2 also has near complete visuals which was made through Affinity Designer, the prototype however still runs through Figma. The final prototype was also tested with users and their feedback was saved for further improvement. The tests are explained in more detail in section 4.7.3.

The app starts similar to the original app, as shown in figure 4.21. The donate button has been made larger with final graphics. The choice to make it larger was to incentivize the user to click it to move forward in the app. The menu is similar to v1.1, however a new button that says "Contact us" has been implemented. The button is a Mailto-link that goes to Räddningsmissionen upon their request. The

language choice is represented with both text in both languages and flags corresponding to the language to help a user change language to their spoken language. The "Start over" button have been made red to signal the user that it will make a major change if pressed, since it will start everything over by clearing all fields. The donation history page have changed minimally from v1.1 the only change is the X-button have been updated to match the color palette and overall design.

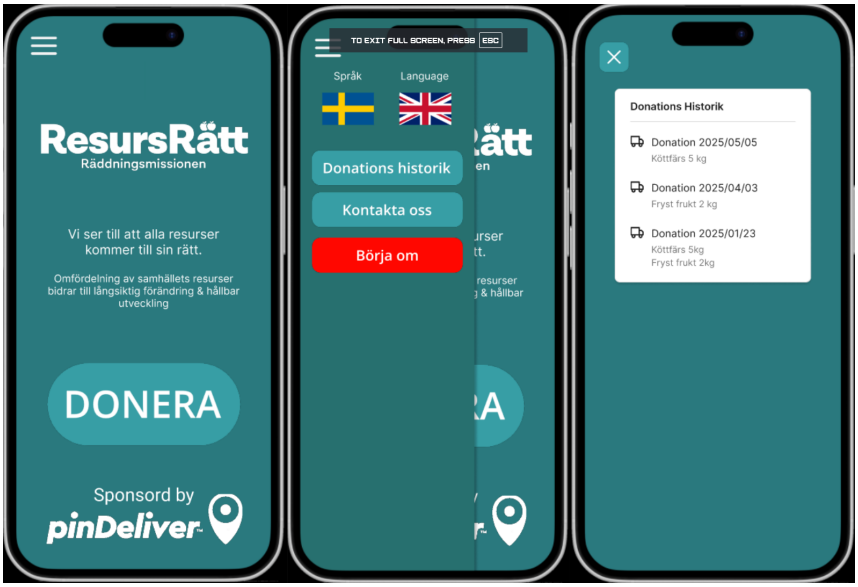


Figure 4.21: The final prototype of the start page, menu and donation history

The donation page has been changed from v1.1 with updated visuals and some of the inputs as shown in figure 4.17. The user now also has an indication on how many steps the app has left through the three dots at the bottom, where each dot gets highlighted when the user reaches the corresponding step. This was added to reassure the user and address the feedback we received, that not knowing how many steps were left was discouraging. The three different head-categories remain as in v1.1 but with updated visuals and larger fonts. The categories are based on Räddningsmissionens usual donations and areas of focus, which are primarily food and clothing. However they also accept other things which gave way to the "Other" button. The buttons now get highlighted on press to indicate which one is currently chosen. If food is chosen the user gets to choose between "Refrigerant", "Frozen Foods" and "Dry Foods". This is crucial information for transport and storage. The user then fills in what it is and its weight, in the weight field the user can only input number so that the user cannot input incorrectly. The weight tells Räddningsmissionen how much they are donating which helps them plan their shipments so that they can send optimal trucks to the location. The best before date and expiration date are so called "date scrollers" that the user can easily choose the correct date and cannot input with the wrong formatting.

The "i"-button has improved graphics to increase visibility, which in turn incentivizes the user to click on it. The pop-up is identical to v1.1, however the X-button now have updated graphics to match the overall design. The user can add additional

information if they want just like in the v1.1. The user can now choose between going forward with the arrow or add another item with the plus sign. The symbols have been made bigger to be bore expressive. The list of items works the same way as explained in v1.1. If the user instead chose clothes, the app asks what item the user wants to donate and unlike v1.1 where the user inputted a set amount the user instead has to describe the volume. The user gets examples how it can be inputted and also an information button have been added to further explain how the user should describe it. Finally the user can fill in additional information if they want and then either add an item or move forward in the app. If other is chosen they input exactly like clothes unlike v1.1 where the user got to choose their own prefix for the quantity. Clothes and other are separate even though they are identical to make the user understand the app is primarily focus on food and clothes not only food.

Figure 4.22: The final prototype of the donation information page

In figure 4.23, the final prototype of the delivery page is shown. From v1.1 not much has changed except for updated graphics and the addition of step tracker which on this page changes to step two of three. The choice buttons are side by side since in the current app during the user tests some users thought that the buttons did not relate to each other when they where over respectively under each other.

Figure 4.23: The final prototype of the delivery information page

The contact information page, shown in figure 4.24, has been changed minimally from v1.1. The only change except for updated graphics are the step tracker which here changed to step three out of three.

Figure 4.24: The final prototype of the contact information page

The thank you page, shown in figure 4.25, now have updated visuals compared to

v1.1, however the "Contact us" button is now a mailto link to Räddningsmissionen instead of leading to their website.

Figure 4.25: The final prototype of the the delivery information page

4.7.3 User Tests of v1.2

To evaluate v1.2 some more user tests where conducted. These users got to try the prototype and express their opinions. The prototype can be run through Figma on a mobile phone. The Figma prototype then simulates the app by running through different actions and animations, like an advanced slideshow. It works by creating different slides where different buttons dynamically lead to other slides much like progressing through an app. This is enough for the user to understand how the real app would look like and work. Since you only can warp between different slides, none of the fields work. The users where asked specifically if they understood what to input in each field. To summarize the users opinions: The overall implementation was solid, simple, straightforward, and fast. There where some areas that they thought should be further improved; primarily the plus button to add new items. The first time the users saw the button they did not understand what it did until actually pressed it. The users therefore suggested the button be changed to a text that described what it actually does, for example "Add another item". There where also some other smaller fix ups they suggested such as changing the button "Donation History" to just saying "History" to make it simpler and the user still understand what history it is. The users also suggested that the email the user gets when sending the order should contain contact information if the users have questions. The users also filled in the "User Experience Questionnaire" as they did in the first test which can be used to compare the new version from the current version. The mean result of the questionnaire are displayed in figure 4.26. Keywords to pick from the questionnaire was that the user thought it was understandable, easy to learn, fast, supportive, pleasant, efficient, clear, practical, organized and friendly.

Figure 4.26: The mean result of the User Experience Questionnaire on prototype V1.2. Green is the new result and black is the old result. If they had the same result they are marked green

4.7.4 Implementation

After a satisfying prototype had been created, work on producing a functional product could be finalized. Implementing prototype V1.2 not only meant revising aesthetic attributes from the inherited app, but also restructuring core functionality inside the codebase. A key difference between the inherited app and prototype V1.x

is the ability to donate multiple different items per donation. The inherited implementation can only handle *one* item per donation, and saves each field in separate variables. An example of a variable stored would be DonationName, storing the name of the item to be donated. The old implementation:

```
public string DonationName { get; set; }
```

Would store values directly from an input element in the View:

```
<Label Text="Vad vill du donera?"/>
<Entry Text="{Binding DonationName}" Placeholder="Ex. Bröd, T-shirts" />
```

An attempt to add multiple entries with this framework would instead continually rewrite the same instance-variable, and would therefore only rewrite the same item, and not support an order of more than one. To enable the switch from one item to many: the object DonationItem, which represents all necessary values for a particular item, had to be created- and separated as its own ViewModel. The example of storing DonationName, would then be replaced by:

```
public ObservableCollection<DonationItemViewModel> DonationItems { get; set; }
```

Where DonationItemViewModel.cs would contain all required variables:

```
public string DonationName { get; set; }
public string Weight { get; set; }
...
```

Binding every UI element to its corresponding instance-variable can then easily be achieved with built-in properties. An example for this case would be using:

```
<CollectionView ItemsSource="{Binding DonationItems}">
...
```

Where property ItemSource="{Binding DonationItems}" tells the View to render one UI block *per item* in the DonationItems collection. Placing all visual elements related to donation items inside this field then make it scalable for multiple objects.

5

Result

This chapter compares the final implementation to previous prototypes and the inherited codebase, as well as outlines sources of errors.

5.1 Final prototype

During the development of the app's many prototypes each suggested feature and/or change was also considered in the context of its ease-of-implementation. This ensured that the feasibility from a programming perspective was considered before any additions were added to the prototypes. Features such as a searchable item-database, and an option for pictures were therefore ruled out early due to either their complexity to implement or incompatibility with parts of the inherited framework that could not be changed.

The final prototype is a pseudo-working implementation of V1.2 in code (which we will refer to as V1.2.1).

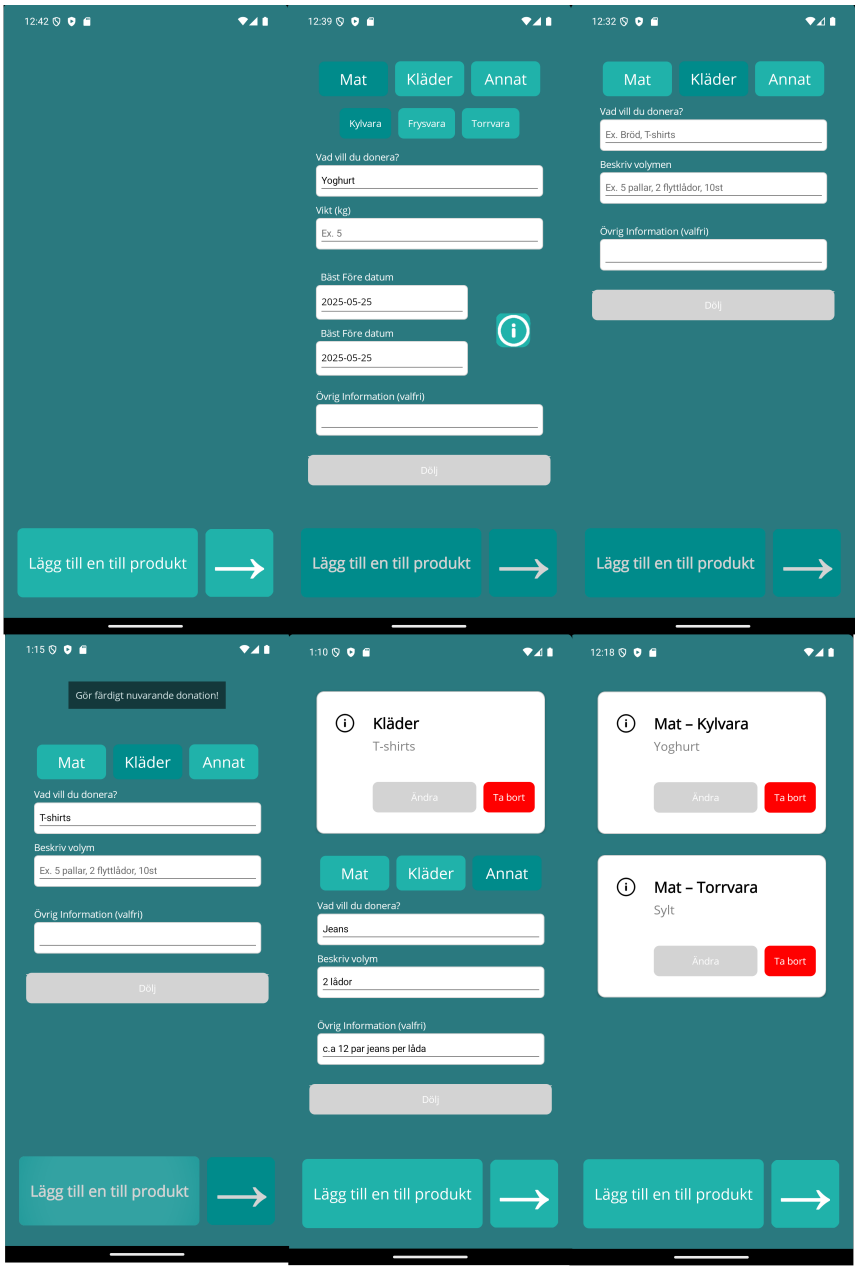


Figure 5.1: Images demonstrating use of the 'Donation'-page from V1.2.1

The final implementation stayed consistent with earlier prototypes in most facets. Visual elements show minor diversions from earlier Figma prototypes; but the placements and intended functionality of the UI elements remain closely similar. Since the donation page had the most structural changes of any page it will be explained first. Notable changes in elements from v1.2 to v1.2.1 in the donation page are the addition of a 'Hide'-button for each item. The existance of this element was strongly implied by v1.2's functionality, though it was forgotten in the prototype. The text for the bottommost buttons was also changed. This is due to additional feedback received on prototype v1.2 while v1.2.1 was in production; the icon "+" was not clear enough in its functionality (as refered to in 4.7.3.
An *omission* made due to technical limitations was excluding custom example-text

for the three different categories of items (the text "Ex. Bröd, T-shirts" in Figure 5.1). Due to the structure of XAML, changing the example-text would require changing the element itself; or more practically creating three separate entry-fields for the name of the item, where only one would be visible at a time depending on which category was chosen. The reason this option is undesirable is that the input-contents in the field would, in the eyes of the user, change sporadically when switching between the three categories. It would also cause issues when the order was submitted, as the variable for donation name would be declared three different times.
Final omissions include the lack of a "select options" button (in the top-left corner, see Figures 4.21-4.25) and progress tracker. There are no technical limitations that prevent these features from being added in future versions, however time constraints disallowed them to be fully completed in prototype v1.2.1

5.2 Technical Challenges

Compared to the inherited codebase, no other page needed as many structural changes as the donation page. The pages for company info (the Company page) and contacts (the Contact page) saw minor functional changes, and primarily needed visual correction to match the design of prototype v1.2. The remaining page-specific logical changes done in the app were the following:
The Company page was re-structured similarly to the Donation page; where a selection from two choice buttons would limit the user to see only the fields necessary, and simplify the process. The code required for this change was designed based on both the new Donation page and the old code existing in the Company page, and thus very to implement.

We also remade the logic for all checks in the app. The inherited code contained many 'fail-safes' to prevent users from progressing through the app without completing necessary steps. Examples include needing to fill in all fields for the donation before continuing, and checking submitted the zip-code and city through a database to ensure they matched. These, along with other checks, were revised to match the new modular code-structure. For example, the previous check on the Donation page consisted of an if-statement-chain that prevented progress until all conditions were met. The new system uses a global IsValid-boolean, that each individual item has the ability to disable. Only when all items present are complete does IsValid boolean return true, and this is used not only as a fail-safe but also for visual elements like highlighting button colors.
After a donation was successful the user would reach the "Thank you-page". This page was cleaned up to remove unnecessary clutter and the options to contact ResursRätt was redirected to modern endpoints.

Some of the most challenging parts of the final implementation include:
The handling of multiple types of DonationItems in general: which meant not every item could be treated the same/be expected to contain the same fields, which required attention to detail when redesigning relevant functions (to make sure they

worked correctly with all item types).

Some inherited code proved difficult to adapt and continue working on. Particularily the naming convention of the original code used very short and inprecise variable-names, which resulted in confusion over the name in-use critical variables and the name of static elements. This was compounded by every line of text in the app being saved as an individual variable, to be change by the language feature (that was yet to be implemented).

Unforeseen circumstances resulted in the project receiving much less allotted time to be worked on than initially intended. From a development-perspective this put pressure on getting functionality to work before the deadline- rather than focus on good coding practices.

Large issues with updating version-dependencies compounded the time difficulty. Getting the code integrated in a working IDE-environment with every dependency updated to the latest versions was an enormous time-sink. Unclear documentation and cluttered error lists meant that stray version miss-matches could take multiple hours to sort out.

5.3 Sources of errors

The following paragraphs are accounts of potential sources of errors which could have created a incorrect result in our user studies.

One source of error would be that not enough users where interviewed, which could mean that some smaller problems where not identified. Another source of error is a lack of adequate company feedback, which could result in missing possible features that companies could want. Many companies where contacted, however only two of the twenty companies responded, which is a ten percent response rate. With more time more companies could be contacted which would have provided more input from said companies.

When comparing the user tests from the old- and the new app, it could potentially have a somewhat incorrect picture. In the first test seven users were interviewed and in the test for the new app only three users were interviewed, where two had already tried the original and one was new. With more time more users could have gotten to try both versions which could have created a more fair comparison.

In this project mainly students where interviewed except for the company interviews. This could have missed problems, however in the interviews with the companies they said that very different kind of people could use it. The key aspects the companies presented was that anyone no matter how much experience they have with using the app or company insight, they should understand and be able to easily fill in the app. Thus using students who have no insight in the project was ideal and many where willing to try the app. Other people than students where asked but rejected

the proposal due to lack of time.

6
Conclusion

A new app where multiple items that are created easier through different categories has been made. The app focuses on asking the right questions at the right time and to guide the user in a supporting manner. The results have improved a lot from the original version of the app and are therefore satisfactory.

In the beginning of the project three research questions where presented. They where:

- What problems do the users encounter during the current process?
- What needs to be improved and what is missing with the current app?
- How do we get more users to the app?

The first question was critical to answer, the user tests identified many problems and each interview highlighted new or the biggest problems. If there was more time more user tests could be done to identify even more problems. All identified problems are displayed in table 4.1. As many of these problems as possible have been tried to be solved in the new version, however further testing would be needed to ascertain that the problems are solved.
The second question was also answered based on the interviews of the users and companies involved in the project. There were many different ideas about missing functions, all suggested functions are shown in table 4.2. All functions worth considering could not be implemented. As mentioned in the previous chapter, some suggested features were not implemented because of the limited time and others because of technical limitations.
The third and final question has not been prioritized to answer since it does not involve the creation of the app. However some questions where asked of the users such as if the app would be more attractive if it was web-based instead, which resulted in a mix of answers some for and others against or somewhere in between. Consensus found that on repeat usage an app is preferential, but for single-time users a website could have more benefits. Other aspects such as how Räddningsmissionen does their marketing have not been examined due to time constraints. However general feedback such as regularly social-media posts and running targeting ads are expected to further increase the apps visibility. Another suggested idea would be to contact companies directly, either through mail, or larger networking events or gatherings.

Regarding the sustainability aspect, this project have created an app that at its core is intended to treasure and collect every available resource to then sell at a low price to people in need. Food that would have been thrown out or clothes that would be discarded can with this app instead be used, which saves the environment

while also helping people.

To summarize the research questions have been answered however more answers could be out there. It would also be interesting to do more tests on the new version to make further improvements. The goal of improving the user experience, have been met in various areas as shown by comparing figure 4.11 and figure 4.26. The app has become more valuable, pleasing, pleasant, secure, efficient and practical, while retaining other aspects to a desired degree.

Bibliography

[1] M. Schrepp, "User Experience Questionnaire Handbook. All you need to know to apply the UEQ successfully in your projects.", User Experience Questionnaire 2023. [Online]. Available: https://www.ueq-online.org/Material/Handbook.pdf

[2] J. M. Bonde, A. Mallalieu, M. Panarotto, O. Isaksson, L. Almefelt, and J. Malmqvist, "'Morpheus: The development and evaluation of a software tool for morphological matrices,", Chalmers University of Technology, 2022. [Online]. Available: https://research.chalmers.se/en/publication/533557

[3] E. Hulthén, PPU196 Produktutvecklingsprojekt, "Steg 5, skapa vinnande koncept", Chalmers University of Technology, 2024. [Lecture]

[4] O. Rexfelt, PPU032 Användarstudier - Att förstå användaren och dess krav, "Observationsbaserade metoder", Chalmers University of Technology, 2024. [Lecture]

[5] Miro, "About Miro", 2025. [Online]. Available: https://miro.com/about/ (retrieved: 2025-06-04).

[6] Figma, "About Us | Figma", 2025. [Online]. Available: https://www.figma.com/about/ (retrieved: 2025-06-04).

[7] Affinity, "Designer", 2025. [Online]. Available: https://affinity.serif.com/en-us/designer/ (retrieved: 2025-06-04).

[8] Microsoft, ".NET Multi-platform App UI", 2025. [Online]. Available: https://dotnet.microsoft.com/en-us/apps/maui (retrieved: 2025-06-04).

[9] Microsoft, "Overview - A tour of C# language", 2025. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview (retrieved: 2025-06-04).

[10] Microsoft, "XAML Overview", 2025. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/ (retrieved: 2025-06-04).

[11] Microsoft, "Visual Studio 2022", 2025. [Online]. Available: https://visualstudio.microsoft.com/vs/ (retrieved: 2025-06-04).

[12] Google, "Meet Android Studio", 2025. [Online]. Available: https://developer.android.com/studio/intro (retreived: 2025-06-04).

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige
www.chalmers.se

CHALMERS