



# HandleNett

by the group consisting of:

Aravinth Sivallingam - [aravinta@ifi.uio.no](mailto:aravinta@ifi.uio.no), 2. year student of Programming and Networks master

Mahasty Assi - [mahastya@ifi.uio.no](mailto:mahastya@ifi.uio.no) , 1. year student of Programming and Networks master

Mazahir Hashimli - [mazahirh@ulrik.uio.no](mailto:mazahirh@ulrik.uio.no) 1. year student of Design, Use and Interaction master

Uy Tran - [uqtran@ifi.uio.no](mailto:uqtran@ifi.uio.no) , 2. year student of Programming and Networks master

Saliq Dar - [saliqnd@ifi.uio.no](mailto:saliqnd@ifi.uio.no) 1. year student of Design, Use and Interaction master

# Index

<b>1. Introduction</b>	<b>3</b>
1.1 Idea	3
1.2 Motive	3
1.3 Research Questions	4
<b>2. Theory</b>	<b>5</b>
2.1 Governance of flexible mobile service platforms [1]	5
2.2 Expanding the 'Mobility' Concept [2]	7
2.3 Principles of Interaction Design	8
2.4 Universal Design	11
<b>3. Development process</b>	<b>12</b>
3.1 Research and Data gathering	12
3.1.1 Survey	12
3.1.2 Interviews	12
3.1.3 User testing	13
3.1.4 Use Cases	13
3.2 Prototyping	15
3.3 Technology	16
3.3.1 Technologies we have used	16
3.3.2 Technologies we would like to use	17
<b>4. Results</b>	<b>18</b>
4.1 The survey	18
4.2 Interview results	19
4.3 User Test Observation results	20
4.3.1 Improvement Suggestions	20
4.4 Use Case results	21
<b>5. Reflection</b>	<b>22</b>
Future ideas for the project	22
<b>6. Resources</b>	<b>24</b>

# 1. Introduction

## 1.1 Idea

Imagine being able to shop for your groceries whenever and wherever, by setting a time and place, and simply picking it up when it is convenient for you. This was the idea our group wanted to focus and experiment on, essentially an app that makes it possible for shop workers to gather the user's groceries based on a defined shopping list made by the user, and then have the user come get it. After some investigation and further discussion the group decided to steer the idea into a more specific direction, where the focus would be that the community could help elders shop for groceries by viewing their shopping lists via the app and decide to help them by gathering their groceries and delivering them to the shoppers home. This gives a more social aspect to the app without involving the businesses too much when it comes to the aspect of having enough workforce.

## 1.2 Motive

The motivation to design and develop an app like this is to make grocery shopping a more effective process. A lot of customers tend to spend too much time in the grocery store, either bringing along a shopping list with them to remind them what they need to buy or wandering around to find whatever it is that they need. With the app the customers may order their groceries while at home, and look around for what they need to order, saving the orders in shopping lists for future use. It would also be economically beneficial for the app users, since they won't be spending that precious time in the store, getting trapped in advertisements, or buying groceries that they didn't intend to in the beginning.

Now since we wanted to focus on an older age group (meaning >50) our motive shifted to wanting to evolve the app as a means to help people living in certain situations where it is difficult for them to shop by themselves. Whether they have health problems that stop them from physically getting to the store, or that they would appreciate some help from others in these daily tasks, they could use the app for such purposes. Imagine

what it could do for a community if teenagers were encouraged to help elders in their area by doing a small favour like getting someone their groceries. This would also benefit families in general with children, but let's focus on one thing at a time.

The app would have two different layouts depending on what kind of role the user of the app has. We were pretty motivated to make the app universally useable. For example for the role of "shopper" we want to make grocery shopping accessible for not only our user group, but also users with disabilities. Considering that smartphones have a wide range of support for people with sight-related problems, the app could be made in a way that it would support voice-over and give both visually impaired people and elderly more independence when it comes to grocery shopping.

With apps that already exist that enable users to shop online and have their food delivered to their home, it's not a stretch to imagine of a future where fewer and fewer people would want to "waste time" going down to the grocery store, so having an app that encourages them to use their local facilities seemed like a good idea.

### 1.3 Research Questions

HandleNett has a lot of issues and research questions that need to be considered and well researched. In order to not be overwhelmed, we decided to focus mainly on finding our user group and see how the app would benefit them specifically. While handleNett is meant to be used by all types of people, we wanted a challenge and decided to focus on the elderly. Making an app with them in mind means that other users would also be able to use it. But *with* that comes a lot of consequences, as very few of us have worked with elderly people in our previous projects. It was hard to say if they would even use our product, as older generations are often very keen on social interaction, and maybe it is a *positive experience* for them to roam around the grocery store or local mall as they shop. We will however have data gathered from people aged >30, as they are people who use mobile applications on a daily basis, and it will be relevant for them in the future to use an app like handleNett.

The next step would be to find out what functionalities are best to have in an app like this. Intertwined with that question is what kind of platform we want to use, and more

about this will be discussed in the theory section of the report. With a user group of elderly people we need to make sure the app is easy to use, and also have consideration for the different problems they might face when it comes to *using* apps on their phones. On the other hand there will be users who actually shop the items themselves, and they need an interface of their own for viewing available shopping lists in their area.

Concrete research questions will be presented and acted upon during the “Research” part of the Development Process point. Lastly it’s important to mention that safety, security and reliability are all factors that needs to be ensured when developing this application, as the app will contain personal information about the user, and have the user’s location shared. Other users, being the role of grocery “getters”, will be able to view the grocery lists and addresses (should they choose to accept a grocery list), so we need to ensure that the information that is shared is safe to share.

## 2. Theory

### 2.1 Governance of flexible mobile service platforms <sup>[1]</sup>

Mobile applications are developed on and used on different types of platforms, and this article discusses how the different aspects of mobile development requires flexibility from said platforms. Based on how flexible you want to make an application, one has to consider three actors; the platform provider, the service provider, and the end user. Finding a platform with flexibility that perfectly satisfies the service provider AND the end user is shown to be difficult at the time the article was written.

The article gives a good description of what one might consider to be a platform; *“A platform may refer to a hardware configuration, an operating system, a software framework or any other common entity on which a number of associated components or services run” - P. Ballon(2009)*

A *flexible* platform means giving specific accessibility to both the user and the service provider without devaluing the platform business wise. Choosing the different types of

mobile platforms will have major implication on all the actors related to the platform, as there is an element of competition in the business world and having too much flexibility will reduce that competitiveness.

The way this article is relevant to HandleNett is based on the fact that we want to make it globally accessible without too much cost. If we were to develop this as a mobile app with native code, we could face the problem of having to duplicate code in order for the app to run on all the different types of mobile operating systems. The opposite to this would be to make a pure web-based application that could run on a browser but make it responsive. Then it doesn't matter what platform it is used on, as the app scales itself based on the size of the screen. In the article this would go under service-provider-centric platforms, giving more flexibility to the service providers but limiting the freedom of end users when it comes to what they can choose to share with the service provider. We mentioned that there needs to be a security aspect when users share their location, and service-centric platforms tend to be able to control the data of the end users, so this brings both a pro and con. The pro would be that it becomes easier to control the "getters" of the groceries and making sure they are legitimate, BUT it limits the flexibility of the end user and might make them uneasy as they do not know how their data is handled. It would be all under the hood for them.

"Governance of flexible mobile service platforms" was published in 2011, and a lot has changed since then. During recent years frameworks like React and Flutter<sup>[5]</sup> have made it possible to write code regardless of what native platform you work on. The service provider will have the freedom to make apps without thinking about duplicating code to fit certain platforms, which saves both time and money, and it also gives the end-users the freedom to get the same app on whatever mobile platform they use. This would be the optimal solution for the actual development of handleNett.

## 2.2 Expanding the 'Mobility' Concept <sup>[2]</sup>

This article focuses on mobility and how our daily lives and lifestyle have changed with the arrival of powerful information and communication technologies (ICTs). We are more

mobile than before, as in, independent from geographical constraints. The article discusses three types; spatial, temporal and contextual mobility, but we will also look at [those other mobility types].

Spatial mobility defines that ICTs have made our life much easier by making us possible to connect with each other no matter where we are situated. Mobile communication technologies such as mobile phones and etc. has energized human nomadicity in urban life. But it is not only the humans who move, objects and symbols and images also have mobility, greatly exemplified by the internet. “Mobility of space” was very interesting as it describes the situation of a user going to the store without actually physically being in the store.

Temporal mobility influences our society and activities where by using technology we are saving time. By using our handleNett, users will save time, instead of going to stores and using time to find a product, they can easily select or search for the product and put it in a shopping list. It not only saves time for the client, but it also makes it easier for client to buy the products they want. The article describes two types situations; *monochronicity* and *polychronicity*. Monochronicity describes people who structurally plan their day and work, whilst polychronicity are quite the opposite, and by introducing new technology that made their work environment more active and temporal, the monochronicity increased. In general, we want nettHandel users to also adopt this attitude, in hopes that introducing an app like this will enable multitasking for an event that usually takes much more space in time in a day.

Contextuality is a concept of being aware of the context in which you are mobile. People use technologies differently depending on the context of their situation. Context awareness is a term often related to developing mobile applications, and has been used and expanded since this article was written in 2002. “Governance of flexible mobile service platforms” mentions that:

*“Mobile services should not only replicate the fixed Internet but may capitalize on novel value drivers like context awareness, personalization and mobility.”*

(Reuver, Bouwman, Prieto and Visser, 2011)

The different roles provide different contexts for handleNett's users, how the users interact with the app gives information about how the app can be improved.

## 2.3 Principles of Interaction Design

When developing apps such as HandleNett it is necessary to be aware of simplicity, visibility, accountability, awareness and feedback. These principles do not only make the design process easier for us, but it also puts the users needs first. Our user group is people above the age of 50, so we need to make the most commonly used task in the app easy, communicating with the user's language and making sure that the user is aware of each step.

### Simplicity

When considering the design for how to choose a grocery store we looked at a map based selector and a list based selector, both with their pros and cons. The benefit of the list based prototype is exactly that it's more simple when comparing it to the map based prototype. While the map prototype can be highly accurate, it could appear too complex for an elderly user who doesn't spend too much time navigating map apps like Google Maps, or has difficulties with the fine motor skills required to navigate a map on a phone screen. The downside of the list based selector is that it's lacking accuracy since it only lists grocery shops nearby of one's home. Therefore we can argue that we want to keep both ways, and giving the user the option to switch.

This is just one example of the challenges we face when designing such an app. An even more difficult question is how we are going to structure the function that creates a shopping list, which is the feature that the user is likely to spend most of their time on. It is crucial to minimize time spent using the app by designing the feature in such a way so that the app won't be a struggle to learn by the user.

### Visibility



We assume that older people might have difficulty with complex apps, which is why all the needed options and materials in a task should be as visible as possible. That means tasks should be visible for users without distracting them with extraneous or redundant information. Having a design with functions that are distracting, confusing and overwhelming and unnecessary will only help the user lose their train of thought and even get frustrated and annoyed with the app.

In our app visibility plays big role. For example, in the app you can order your groceries and pick it on the store. To make this happen, first you need to create a new grocery list and later on chose the products you want to buy, and add it to the list and pay for the product. All these steps are options and materials that should be visible for the user so they know how to manage the task. That is why they shouldn't be confused with unnecessary information.

### Accountability

There is a need to understand the user's actions in the app. When we make a app accountable we make the actions and lack of actions obvious for the users. This can reduce and even prevent mistakes that users can make. We don't want the user to pick an item and go through a long process only to find out they didn't want it after all. We shouldn't give a room for this kind of errors in handleNett. It should be obvious for the user what kind of product they have chosen, and what product they are paying for. If the information is clear and accurate, the responsibility is on the user when the payment is done.

### Awareness

Awareness makes it clear for the users to know where they find themself in the application. That means when we are navigating in the app sometimes we are jumping from the task to task, and later we don't even remember what was our real goal to do. Facebook app can be example for this situation. If you want to search for someone, it is easy to be distracted by the newsfeed, notification, messages. Because we have many tasks just in front of our eyes, we can easily forget about our main task.

That is unacceptable situation in our app. People in our user group can easily lose themselves in the app. That is why the app should be simple to navigate when it comes to the simple tasks. There shouldn't be too many steps to manage the task, so they don't lose awareness of their location.

### Feedback

Feedback notifies users if an action is not properly managed or needs to be managed. It keeps users informed of their actions or interpretations. Good feedback should explain the error or message in a very simple way and also propose a solution. For example in handleNett, if the user has created a grocery list and haven't payed for it yet, they should get a notification like "You haven't payed for your grocery list. Pay Now?"

### Usability

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

Learnability is one of five component in usability where we check how easy it is for the user to accomplish basic tasks the first time they use the design. The design is good if we give a task to user to "buy 1kg pears" via handleNett and the user will easily understands what needs to be done. Efficiency comes after the application has been used once or several times. Here we check how quickly users can perform the tasks after they have learned/gotten used to the design. Often users are more effective doing tasks with less or no mistakes after they have learned the design.

Memorability checks how easily the user can reestablish proficiency. When we update the app still it should be clear for the user to use the app as effective as before without making mistakes. That is why user should easily see what has been updated or not

updated. For instance if the app has two payment methods as pay with paypal or vipps, if after update there is third payment method is added user should notice it easily.

Error checking also should be done. It is necessary to know how many errors the users make how severe are this errors and how easily they can recover from this errors. Are errors fixable or they can create much more problem for future use. Satisfaction checks how pleasant is it to use the design. Observation of use of the app is a good way to see all this five component are respected. We have done some observation on our prototype to check all that.

## 2.4 Universal Design

For handleNett to have a universal design we need to make sure that what needs to be captioned and described should be so. Buttons with certain types of text should be descriptive enough so that applications like VoiceOver<sup>[6]</sup> and captioning technologies can give proper feedback to users who need it. In his presentation, Trenton Schulz<sup>[7]</sup> talks about how web applications can be slow and in general not optimal when designing universal apps, which makes the development of frameworks like React crucial in order to make native apps, but at the same time be independent from writing code in native languages.

Colour usage is an element that should be carefully considered, we do not want to use colours that are difficult to distinguish for those who are colourblind, or colours that can trigger epilepsy. When it comes to motor skills the hardest parts in the app would be the registering of a user, search, and if they choose it, navigating the map. There could be an issue with overwhelming information, if a grocery store is large they will most likely have a multitude of items in one category, and users will not want to waste time looking through each and every one. HandleNett would gain from having a powerful search function with, perhaps, the option to speak and find what the users are looking for.

## 3. Development process

### 3.1 Research and Data gathering

#### 3.1.1 Survey

A survey is an excellent data gathering method to get overview over project relevance and user groups with the help of statistics, and a general way to find out whether the app covers any of the users' needs. With the help of a survey, we will be given an idea about what potential users think about the concept of handleNett. The survey was obviously anonymous, so that the users could feel more comfortable giving their honest opinions.

For the survey we had both closed and open-ended questions, where the goal was to get information on whether handleNett is a relevant application, and eventually what we can change to make our development process better. When the survey was first shared with people, those who answered were of all age groups and had very varying replies on almost every question

#### 3.1.2 Interviews

We used observation as a research method as well, visiting a large shopping mall and finding at what times our user group would be most active. During our observation we decided to ask some older people whether they had time for a short interview, to get some qualitative data as well, since we had paper prototypes that could be shown for getting feedback on design and asking open questions to see what our potential end users care about.

#### 3.1.3 User testing

Having a prototype with sufficient functionality designed in it we could test the flow of an action with potential end users, as having a physical example to work with gives very valuable information about the eventual flaws in design and functionality.

### 3.1.4 Use Cases

Use cases are a simple way of describing the flow of an action in order to think through what could go right and also what could go wrong. For our use-cases we also created some simple personas for the different tasks they were supposed to do. This method was excellent for finding flaws in our processes that we hadn't thought of before.

Actor	Task level goal
Heidi	Wants to make a order for nearest grocery and pick up items at specific time.
Walid	Wants to help someone get their groceries.
Nina	Wants to register a account

**Heidi** is 65 years old woman and she uses a wheelchair in daily life. She lives alone in her apartment. Heidi has friends as neighbors, who are around the same age as her. Everyone in the neighbourhood try to help each other and they share most of their time together. Heidi is uses a smart phone, and she got recommended to try "handlenett" to purchase grocery items from one of her friends.

#### Use case:

1. Heidi opens the app "handlenett"
2. She chooses a time to pick up the groceries
3. Then she choose the nearest shop from the map, that is available on that time
4. She chooses items from different categories and adds items to her shopping cart
5. Heidi goes to cart and confirms the goods
6. She gets options to "save shopping list" or "pay the order"
  - a. She chooses "pay the order", and pay through the vipps system
  - b. The system sends Heidi a confirmation mail and the reference number for the order

**Goal achieved:** Heidi books her items

#### Deviation and exception handling:

3. At the given time, no shop is available to take the order.
  - The system prompts the user to select a different time.
4. Specified item does not exist or is out of stock
  - The system offers the user to choose other options of similar goods.

---

**Walid** is a young man who lives close to Heidi, he goes to a high school a few blocks away from where she lives, and is familiar with the tenants of Heidi's building as he sees them around in

the park. He uses “handleNett” because his mom is often busy with both work and housework, so she makes him get the groceries she lists up in the app. On his way home from work he reflexively decides to check the app to see if his mom has put out any new shopping lists, but instead sees a new list popping up close to his location.

**Use case pre-condition:** handleNett is open, user is on “check shopping lists” section, user also has a setting on for “bringing groceries to address”

**Use case:**

1. Walid selects the new shopping list that has appeared close to his location
2. The system shows amount of items, which store they are requested from, and the area the groceries needs to be delivered. The system also shows a small icon signifying that the requester is disabled.
3. Walid presses a button to accept getting the requesters groceries.
4. The system displays more detailed information about the requester, as well as showing the grocery list
5. The system also alerts Heidi that someone has picked her shopping request, shows her that person’s information, and informs that this person will bring the groceries to her address.

**Goal achieved:** Heidi has gotten help from a local with her grocery shopping

**Deviation and exception handling:**

3. Walid chooses to not accept the request
  - The system removes the shopping list from his list (temporarily)

---

**Nina** is 50 years old and also wants to use “handleNett” after having seen Heidi use it for her purpose. She need to register as a customer on the “handleNett” system. She decides to use an Iphone to register.

**Use case:**

1. Nina opens the app “handlenett”
2. Nina selects new account option.
3. She gets an option to register with a facebook or google account.
  - a. She chooses to register through her google account
4. Nina gets the home page of “HandleNett” with a message; “you are registered”.

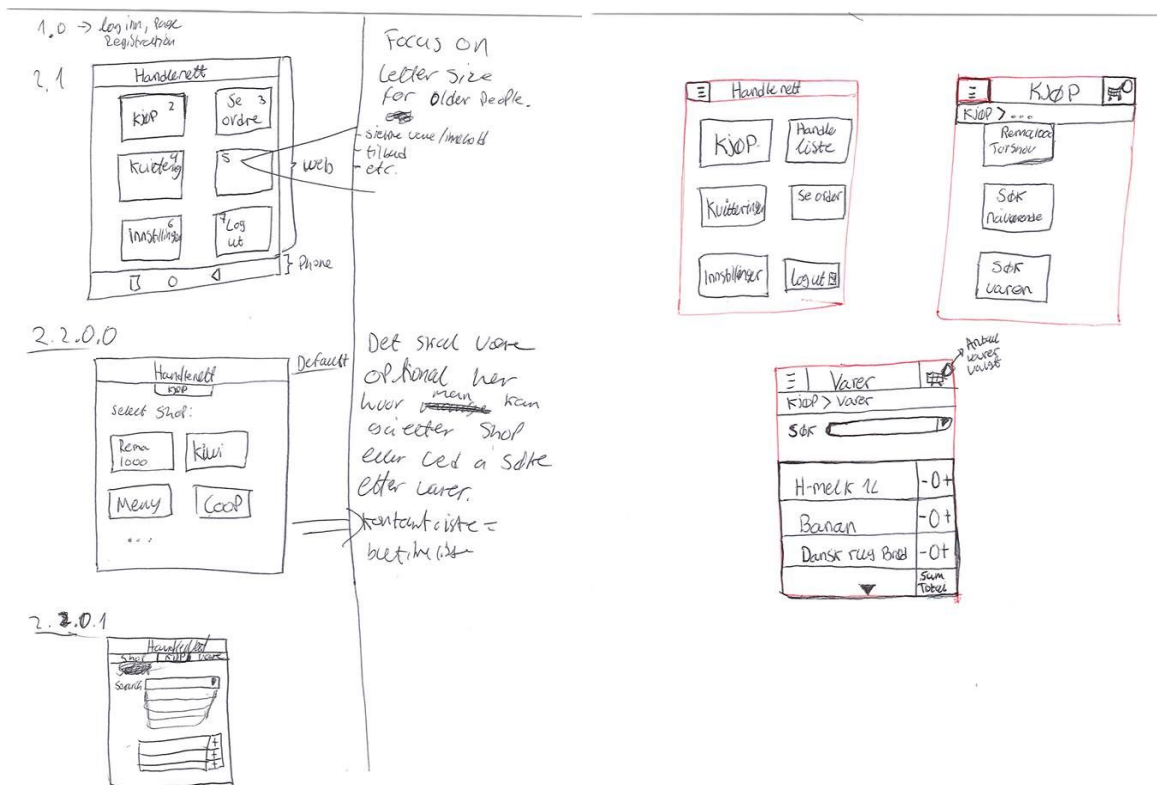
**Goal achieved** - Nina manages to register a new account.

**Deviation and exception handling:**

2. Nina is already logged in
- 3a. Nina doesn't have a google account
  - System gives facebook option
  - System gives an option to fill in all the required fields to register manually

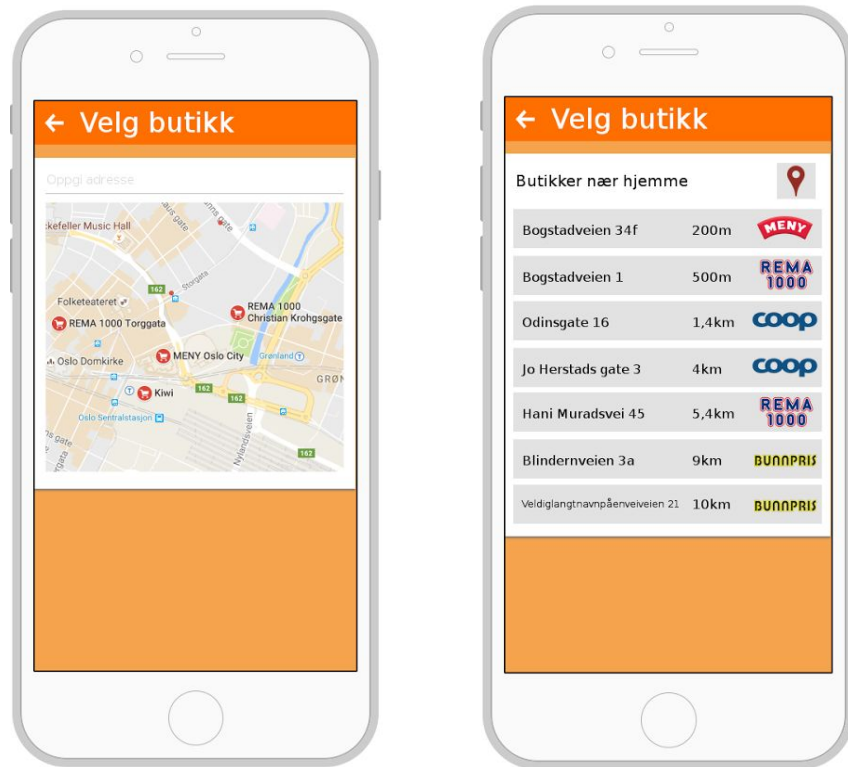
## 3.2 Prototyping

For the design we made a few sketches and talked about how to best design the layout of the app, and what colours to use and how to best make the app as intuitive as possible. Low fidelity prototyping was used to sketch out ideas and brainstorm.



A couple of the sketches made for the low-fidelity prototyping of the user interface.

Having an idea about how the flow of the app would go the design process of a mid-fidelity prototype could begin. The prototype was made with its main functions added and was then printed out so it could be used for user testing.



*Two mockups of the feature that selects which grocery store the consumer is going to pick up the groceries at. The image to the left is a map based selector and the right image is a list based selector.*

### 3.3 Technology

#### 3.3.1 Technologies we have used

*Google's Material Design* is a user interface framework, that has its own guidelines for how to design user interactions and accounts for Universal Design (UD). Since UD is mandatory in Norway for any application developed for the public, this comes in handy. Material Design is also an overall easy to handle framework that is compatible with ReactJS which is also a framework we want to utilize. AngularJS is another popular framework, we may consider since AngularJS version 2 recently released in 24. September 2016<sup>[3]</sup>.

For mid-fidelity mockup sketching, we decided to utilize *Proto.io* because we found it easy to use and it also supported Material Design user interface elements. This made it possible for us to mock prototypes that could be as close to the real app as possible in a fast



and easy way. *Proto.io* also has animation features which allows us to animate the app with different transitions and button triggers. This means that we could almost make the whole app in *Proto.io* being as close as a prototype can be to the real thing. Although animating pages and different transitions in the app is overall a good feature to have in a prototype, we decided not to utilize this feature in our project. This is due to the project's time frame and that we don't want to focus too much on the prototypes.

*Google Forms* was used to create a survey for gathering user data and gauging interest. Google Forms is free and easy to use, and makes it possible to plot the results into an excel document if that is needed<sup>[4]</sup>.

*Facebook* was used, where we shared our survey on both private profiles and a grocery page. Many older people know how to use facebook for communicate with family and friends, though there's not a 100 percent guarantee that all elders are active on social media, especially those with a generation gap too big and a certain amount of contempt towards new technology.

### 3.3.2 Technologies we would like to use

Now that web-browsers are getting stronger and more flexible themselves, we find that we may never need to develop native apps with their native language in the future since, the technical differences between native apps and web-apps becomes trivial. With the dawn of HTML5 we may even develop web-apps that runs offline where the app is stored in the browser cache memory by utilizing the browser service workers. This will allow software developers to develop features in web-apps that may be used without an internet connection. The difficult task is to decide what data to fetch from the servers and what data to fetch from the browser cache and the developers. We may consider running certain features in HandleNett as offline features, for instance, viewing receipts. Though it would mean that receipts would need to be stored locally on the user's phone.

## 4. Results

### 4.1 The survey

Our survey gave us good base data, and because it was posted in an environment with all age groups, the majority of participants were between the ages 19-30. We gathered data *specifically* from our main user group via casual yet selective interviewing, but we also got feedback from people >50 years old through the survey. The most interesting points were:

- What platforms the participants use, as this backs up our wish to make an app to support both Apple and Android phones (and other). There was a mostly even distribution between iOS and Android users, but the windows phones were the minority.
- How much time people spend in the store, to see if the process could be made more effective. We imagine that for families that have a lot to do and for people who have difficulties walking around too much, cutting down on the strolling time in the store will be a plus.
- The responses to our open question "Would you use this app to shop for groceries?". Here we got insight, as people in different situation commented based on their own experiences.

With that last question we noticed a very important point: Some people enjoy the experience of going to the store and taking their time, interacting with the store workers, and as mentioned before, this might apply doubly so for the really elderly generation, especially if they live alone.

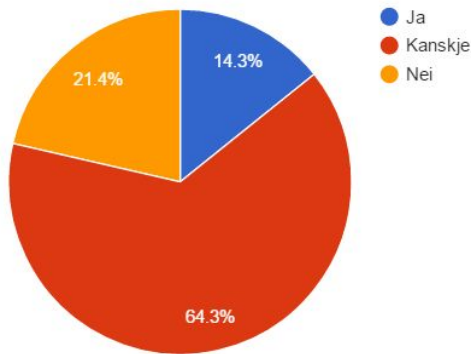
### 4.2 Interview results

To add to our survey, we went to Storo shopping center and had interviews with people over 50 years of age. We got both positive and negative responses from the five interviews we had. Most of the people we interviewed asked us about the quality of fresh products and

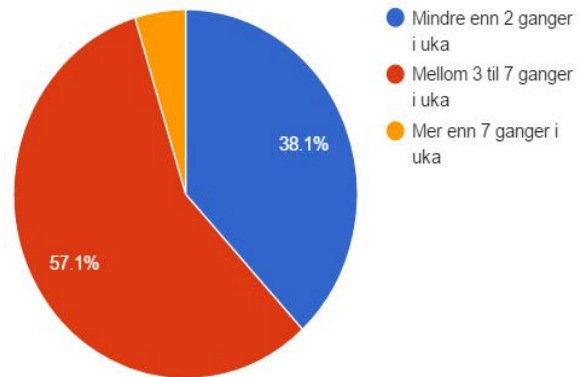
how meat and fish would be packed. For example, some people like green bananas than yellow, or are very specific about the type of fresh meat they want to get. In this situation there would be a trust issue related to the person who gathers the products. We can maybe put a comment line for each product that gets ordered, so the customer can describe what special need they have. The prices also was important point for the customers when it came to online shopping. If the prices are higher in the app, less people will want to use the app according to the answers we got, so the prices should be the same as the store.

We found out that they want also advertisements for goods are on the sale so they can buy the cheapest goods. 80% of the people we interviewed were positive towards the idea of getting groceries for others. They mentioned that this concept would be good for elderly people and families. Women were more positive to the app than men. Mostly the answer to “would you use this app” was weak yes, because they couldn’t visualise that they would use the app every day, but they were positive to use it more if they happened to be sick or busy.

Hadde du brukt "HandleNett" applikasjonen vår for å handle?




Hvor ofte handler du i butikken?



*Picture of pie charts describing the interest in using handleNett and how often people shop in the grocery store, ages >30.*

### 4.3 User Test Observation results

A woman in her late 60's was our first user tester for our application. We had a paper prototype of the app which we used to get information about the usage of the app when a

person would want to buy smoked salmon. When she was on the homepage and wanted to make an order, she couldn't find the option to make a new order. The user felt confused and overwhelmed, and since she was the first user we tested, she got a hint for where she could navigate further. The *new order* function was hidden under a hamburger menu (  ). She managed to navigate through all the parts of the application. The confusing part for her was where she adds an item to the cart and then suddenly she's on the category-screen, where she gets the different categories again. No feedback was given about what happened to the item she wanted to buy. In that situation she felt confused and didn't know what had just happened. The user lost focus of the task because of no feedback came from the system, so we gave her another hint as we wanted to get as much information as possible. The rest of the test went smoothly and she felt confident for completing the task.

Further tests on different users showed that most of them stopped on the same areas as our first tester,

#### 4.3.1 Improvement Suggestions



*Global navigation bar*

After we had gone through user testing, the biggest issue for the user was to find the “new shopping list” option. Since there are only two functions under the hamburger menu, it is currently unnecessary to have it and should instead be replaced with global navigation. With the help of a global navigation bar, it will be much easier for the user to navigate to different part of the application, and the functions are all visible. It is crucial that user doesn't feel confused and lost.



## *Breadcrumb*

Breadcrumb is another feature that helps user to keep track of their location and it can be used as second option for navigation also. In our first test, user lost track of her location because when the user added an item to the cart, the application didn't confirm that the item had been added to the shopping list in a clear way. To solve this problem, we decided to increase the size of the badge, so that the badge is more visible. Snackbar is a rectangular bar with a message that pops up from bottom of the page and then disappears after few seconds. By adding a snackbar we increase the feedback to the user even further.

### 4.4 Use Case results

The use cases really drove home the point of wanting to make the app more about people helping each other and the grocery stores being a third party that simply supply the goods. A hurdle would then be making sure that there *are* people in the local community willing to help, because we do not want a scenario where a requester puts out their grocery list and nobody helps them get their groceries. It would therefore be very relevant to have a system of static getters. A person could choose to be the "sponsor" of someone needing help and making sure they they would get their groceries on a regular basis.

We also considered pick-up option, where the "getters" shop for the items but *don't* bring them to the requester's home, instead waiting for the requester to grab the groceries at the time they noted in handleNett. Here, of course, there would be a risk of waiting for someone who ends up not arriving on time at all, or a requester that picks out a shopping list but does not fulfill it. An alarm system could be integrated with the app to remind users when the pick-up point is close for both parties to know when they should be at the store, for either getting items or picking up their groceries.

## 5. Reflection

Mobile technology keeps getting more and more advanced, and us as users continue to depend more on functionalities in order to make our days more efficient. This might not be the case for every generation, as it is right now, there are older people in technologically advanced countries that do not use and do not like to use platforms like smartphones and tablets. As this was our first challenge, we realised that as time moves on, so will technology. With the growth of mobile platforms and applications, technology will often be a few steps ahead of users that are more comfortable with and have experience with older devices. The app won't be interesting for everybody, but it will be very relevant for tight-knit communities that depends on each other and want to help each other, enabling regular human beings to assist in making a day easier for someone who needs it. But the same time as there is dependency on others in order for the app to work, there is still an element of independence and choice connected to those who want to choose their own groceries.

In a world where people comment on how mobile devices have isolated us more from human contact and interaction, making apps that are based on these kinds of social interactions boosts morale for those who help, and proves to older generations that ICTs can also bring people together.

### Future ideas for the project

Since our group decided to focus on the user experience of an older age group in relation to handleNett, we didn't get into all the other functionalities this app could have, as it would take more than four weeks to go through it all and document it as well. Given that the app is meant to be used by large families, a big beneficial aspect was related to how grocery stores often buy more than they sell, and more often than not food is wasted because the expiration date has passed before anyone bought the product and so on. Mass production and excessive waste means a lot wasted energy and resources which obviously affects the

environment. Having statistics from the app when it comes to what users buy and how often, and how much, would give valuable information to each local store for them to know what to order, when, and how much. This would eventually be a long term project as it takes time to gather such data, but it is still an interesting direction the app can take.

## 6. Resources

[1] de Reuver, M., H. Bouwman, G. Prieto, and A. Visser. "Governance of Flexible Mobile Service Platforms." *Futures* 43, no. 9 (2011): 979–85.

[2] Masao Kakihara & Carsten Sorensen: *Expanding the 'Mobility' Concept*, 2001. SIGGROUP Bulletin December 2001Nol 22, No.3.

[3] <https://en.wikipedia.org/wiki/AngularJS>

[4] The survey made for gathering user data  
[https://docs.google.com/forms/d/e/1FAIpQLSc\\_K0a8g1HAzoo01I-1Y4452kNNA5z4mmY276uqfKVzbp6w/viewform?c=0&w=1](https://docs.google.com/forms/d/e/1FAIpQLSc_K0a8g1HAzoo01I-1Y4452kNNA5z4mmY276uqfKVzbp6w/viewform?c=0&w=1)

[5] React : <https://facebook.github.io/react-native/>  
Flutter: <https://flutter.io/>

[6] Apple description of VoiceOver  
<https://developer.apple.com/accessibility/ios/>

[7] Trenton Schulz: Universal Design & App Development  
<http://www.uio.no/studier/emner/matnat/ifi/INF5261/h16/undervisningsmateriale/Trenton-Universal-Design-Apps-2016-10-18.pdf>