

Titel TBA

A software analysis and -implementation

BILLEDE

*Mathias Vestergaard Rasmussen
Christoffer Carlé Christensen
Kasper Østergaard Helsted
Anders Lykke Matthiassen
Christian Stephansen
Gideon Blegmand*

19 - 12 - 2014

This page intentionally left (almost) blank.

Title:
Titel TBA

Project Period:
Autumn Semester 2014

Project Group:
SW3-ds301e14

Participants:
Mathias Vestergaard Rasmussen
Christoffer Carlé Christensen
Kasper Østergaard Helsted
Anders Lykke Matthiassen
Christian Stephansen
Gideon Blegmand

Supervisor:
TBA

Copies:
8

Page Numbers:
72

Number and Type of Annexes:
3 pages, documentation

Abstract:

This report will focus on food waste. It will discuss the causes of the problem and then suggest a software solution. The report will first discuss the problem with a focus on the causes and possibly solutions to these. This general view will then be used to create a problem statement for the software solution to solve. Afterwards the architecture and interaction design of the solution will be formulated. The architecture is considered with defining the architectural design. The interaction design defines a design language which is a standard of the solutions interaction design. The architectural and interaction design are then used as a basis for the actual solution program. This program is the product of the report. The last part of the report concludes and sets the future guidelines for the program.

Appendix:
Page 70-72

Date of Completion:
19-12-2014

The contents of this report is freely accessible, but publication (with referencing) may only happen under agreement with the authors.

This page intentionally left (almost) blank.

Preface

This report is a result of the hard work of Anders Lykke Matthiassen, Christian Stephansen, Christoffer Carlé Christensen, Gideon Blegmand, Kasper Østergaard Helsted, and Mathias Vestergaard Rasmussen.

The intellectual property rights of this report and all of its contents belong to the members of the project. All material gathered from third party resources will be refereed to according to the IEEE standard.

The source code for the software solution of the project, is open source, under the Beerware license. The code is freely available at <https://github.com/Helstedxd/FoodPlanner-Project>.

The colours for this program, were chosen by using an online tool called Paletton [**paletton**], which suggests a colour theme based on the colours the user choose. The figures made by the report makers, have been created using Visio and Inkscape.

The makers of the report would like to thank participants of the Usability test for their time and constructive critique, as well as the informants for the interviews, for their cooperation. The report makers would also like to thank Rudra Pratap Deb Nath for his help and constructive criticism throughout the project process.

Contents

I	Introduction	1
0.1	Chosen Structure Method	2
II	Problem Analysis	4
1	Information Gathering	5
1.1	Food waste	5
1.2	FDB Report	5
1.3	How to avoid food waste	7
1.4	Foodplan study	8
1.5	Interviews	9
2	System Choice	12
2.1	Situation	12
2.2	Ideas	14
2.3	System	17
3	Problem Domain	19
3.1	Classes and Events	19
3.2	Structure	22
3.3	Priority of criteria	25
4	Application Domain	27
4.1	Usage	27
4.2	Function definitions	30
5	Problem Statement	32
III	Design	33
6	Scenario-based design	34
6.1	Model Development	34
6.2	Prototype Development	40
6.3	Design Development	58
7	Architecture	63
7.1	Component Architecture	63
7.2	Process Architecture	64
7.3	Database Model / Classes	66

7.4 Model View ViewModel	67
7.5 Entity Framework	68
8 Design Conclusion	70
IV Product Development	71
9 Pages	72
9.1 Settings	72
9.2 Inventory	72
9.3 Shopping List	73
9.4 Mealplan	73
9.5 Search	73
9.6 Recipe Page	73
10 Custom User Controls	74
10.1 Ingredient Auto Completion	74
11 ViewModels	75
11.1 Mealplan	75
11.2 Recipe	75
11.3 Inventory	76
11.4 Settings	76
12 Navigator	77
13 Testing	78
13.1 Usability Test	78
V Summary	81
14 Delimitation	82
14.1 Not in program	82
14.2 In program	83
15 Conclusion	85
16 Future Improvements	86
16.1 Improvements	86
16.2 Reflection	86
VI Appendix	88
A Interview answers	89
B PACT Analysis	95
C Personas	101

D Design requirements **104**

E Usability Test Tasks **107**

Part I

Introduction

0.1 Chosen Structure Method

In this section the Waterfall and Iterative method will be described which both are used to structure a project. The concepts of the methods and their strengths and weaknesses will be discussed. This information will be used to support the choice of method. The chosen method and its application in the documentation will be described afterwards.

The waterfall method iterates the project process one time and avoids revision the parts that have already been finished. That means that a part is worked on until it satisfies certain requirements, and then the project members move on to the next part. The iterative method focuses on multiple iterations of the project process and therefore iterating through the project multiple times, expanding the product and documentation each time.

Both methods have their strengths and weaknesses and one method might have a strength that is a weakness for the second method. One of the strengths of the Waterfall method is the consecutive order of progress the structure gives. This gives the project members an overview of how far they are in a specific part or in the overall progress of the project. This overview is obtained as the participants of the project can documentation when a part of the project is finished. Another strength of this method is the ease of setting deadlines for the parts in the documentation and using these deadlines to measure the progress of the project. The consecutive nature of the method can also become a weakness. During the project information can change and this can result in previous parts of the documentation not including or reflecting the new information. This also leads to another weakness which is the question, is the product of the project really complete when the project is done? The project can expand and new information can change, add or subtract requirements for the final product. So it is difficult to conclude that the product is complete when the project is done.

The iterative method has the strength that it is dynamic. Each iteration of the project process allows the addition of new information and changes. This means that this method adapts very well to new information and changes. When using this method it can be difficult to keep an overview because each iteration adds more information. Each iteration also gives a better understanding of the project context, development and requirements of these. This could lead to an endless circle of improving and expanding the project. So when is the project actually done when using the iterative method? This could prove to be difficult to answer, whereas the Waterfall method gives a structure that restricts the multiple iterations and sets a end point.

The Waterfall model gives a good structure with an easy overview of progress. It is also a good choice because the system requirements and the goals that have been set, can be defined after each part of the documentation. This allows for confirmation on the goals and requirements that have been set. The Waterfall method will therefore be used in this project.

When documenting a project using the waterfall method as a structure, it is necessary to write five different parts which covers the project process from start to finish.

The analysis part focuses on getting to know the project in context e.g. to the audience or the environment. This is the part of the documentation that is called problem analysis. It is based on relevant sources e.g. articles, documentations, questionnaires or interviews which satisfies a certain standard of source criticism and relevance. In this project the information is gathered through external sources and interviews. The information is then used to define specific goals and requirements for the following parts of the documentation.

0.1. CHOSEN STRUCTURE METHOD

The design part is used to identify design specifications that is needed in a certain product. In our documentation, the design part focused on the principles learned in the courses "Design and Evaluation of User Interfaces" and "Object Oriented Analysis and Design". In this part of the documentation the interface design is modeled by using scenario- based design. The architectural design is created by using Object Oriented Analysis and Design techniques.

The implementation part is the part in which the software is created. The design specifications is used to make the design, and the needed functionality requirements which was identified through the analysis part is implemented. The outcome of this process in the project is a functional program.

The system test in this part the system is tested to validate if it satisfies the stated requirements and goals in regards to both usability and quality. In the documentation the system is tested to secure that it is of high quality and the usability of the program is also tested.¹

The system in operation is concerned with describing the system when and after it is published. The documentation incorporates this part as a reflection and evaluation of what is written previously as a closure.².

¹Fixme Note: skriv mere præcist nå testing er done

²Fixme Note: beskriv bedre når det er lavet

Part II

Problem Analysis

1. Information Gathering



1.1 Food waste

Every year danish households throws away nearly 237.000 tons of food, that could have been eaten by people. This is estimated to be nearly 16 mio DKK. About 20 % of a danish households foodbudget is wasted by throwing away food that could have been eaten, this means that if there was no food waste at all, 1 million people could be fed, this is nearly 1/5 of the danish people.

The people who throws away most food, are the people living alone. The average person living alone throws away 98.8 kg of food per year, where as a household with 5 people throws away 46.8 kg of food per person per year. It is mostly dairy products, vegetables and bread that gets thrown away, though in december a lot of meat is also thrown away, due to the fact that it is Christmas, where people eat more food.

Since 2007 the danish people have been purchasing 10 % less food, a lot of this food used to end up in the trashcan as food waste. 54 % of the danish people rarely or never uses the food from the night before for later eating, and 69 % does not use leftovers for lunch the day after. Elderly people are better at using leftovers than younger people.¹



20 % of danish people does not make a grocery shopping list before going grocery shopping. 52 % of people does not make a meal plan, 31 % does sometimes and 17 % does it often.

81 % of danish people would use leftovers to avoid food waste, 50 percent would make a mealplan and make a grocery shopping list. 59 % of the consumers believe that the danish food waste is their own fault.

1.2 FDB Report

The following section examines a report published in 2011 by FDB called "Forbrugerne: Vi smider ikke mad ud"[\[madSpild_FDB\]](#).

1.2.1 Method used in the FDB report

The FDB report is a study of how food waste occurs and how it is experienced by the consumers. The study is based on qualitative data collected from six different households. These households are all different from one another e.g. they live in different geographical locations. Data is gathered from the participants by using observations e.g. watching participants shop or prepare food, semi-structured interviews and probing kits. The data is then used to examine the activity patterns and motivation of the participants.

¹Fixme Note: references to bibliography, and/or maybe some graphs to show the numbers?

1.2.2 FDB report: What is food waste and how does it occur.

The FDB report discusses why food waste occurs and how the participants perceive it. There are some subjects that are interesting when learning about food waste.

What is food waste

When the participants were asked how much food they threw out, they typically answered that they threw out as little as possible. There is a clear distinction between waste and non-waste being thrown out. It is acceptable to throw waste out, but not to do the same with non-waste. The waste and non-waste categories varies depending on cooked food products and uncooked food products. Cooked food is considered waste when larger parts, or leftovers of a meal that has not been served on a plate is thrown out. So when the food has been served on a plate it is acceptable to throw it out. Uncooked food is considered waste when the packaging is unopened. It is not considered waste when the uncooked fat or other food parts that are considered non-edible is thrown out.

Cause of food waste

The reasons as to why food is thrown out varies depending on cooked and uncooked products. Uncooked products is thrown out because they are bought in so large quantities that the consumers were not able to eat it prior to expiry. Products that are considered non-edible such as cartilage or bad vegetables are also thrown out. Cooked food products are also likely to be thrown out. Cooked food products are thrown out because the participants prepared more than they could eat in one meal. If the participants estimated that there were enough leftovers to save them for later they would. But often the leftovers were stored in the fridge for 4 days only to be thrown out.

Food waste barriers

The participants might have the will to reduce food waste but there are some barriers that can impact the participants will in a negative way. The value of the meal ingredients can be a barrier. Dinner is often based around the chosen kind of meat because it has more value. This means that all other ingredients on a plate only is supplementary. It is easier to throw away the supplements than the meat because of its value. Therefore supplements represents a larger part of the food waste in the homes.

Making a large delicious and fresh meal is also a way to show appreciation towards guests or household members. This can also act as a barrier because it requires the food to be fresh and in large amounts. Another barrier is quantity discounts. It contradicts with the participants reasoning when they have to decide between buying what is the right economical, or what is right according to food waste. In the situation people often think of the economical aspects, and not on what they will do with the extra food. The participants also expressed a lack of overview of what products they had at home when they were out shopping. Sometimes they bought something that they already had, because they was not sure if they had it at home. This results in an overstocking of a product that is most likely to be thrown out.

Another barrier was planning versus impulse and desire. Participants that shopped regularly, said that they sometimes changed their minds on what they wanted for dinner or that they wanted variation in their meals. This spontaneity and need for variation could result in products being bought for one night and the leftovers being stored and in the end being thrown out. Participants that shopped frequently said that they wanted variation and therefore did not plan dinners for a whole week. This could result in a lack of overview and in the end more food waste. The expiration date can also affect the participants willingness to buy a product. The tolerance of when a product is not edible varies between participants, and when the product can give a smell or a visual sign of decay. The expiration date had less importance. The last barrier that is discussed is what is acceptable as food waste. Some of the barriers that has been mentioned is accepted in some degree by the participants. One of the participants talks about the acceptance of food waste when it comes from a child's plate. The participants says that it is to hard assess how much the child will eat and that it leads to food waste but it is a waste the participant will accept.

1.3 How to avoid food waste

With 20 % of grocery being bought in a Danish household getting thrown away, finding ways to decrease or prevent food waste is essential. When throwing away food, it is not only groceries that could have been eaten, which are going to waste, but also the energy which have been used to grow, make or harvest the product are being wasted as well.

According to The Danish Ministry of the Environment[[madSpild_Notat](#)], one of the initiatives that can be taken against food waste, is to make recipes which incorporate the leftovers, and other uncooked groceries. By doing this, food that might otherwise go bad, will be used instead of being thrown away.

Another initiative that can be taken according to The Danish Ministry of the Environment, is to sell quantity discount, with the option to receive some of the groceries later. An example would be if one cucumber is sold for 10 DKK, while two is sold for 15 DKK, some people would choose to buy two cucumbers because of the quantity discount, even if they just needed one at the moment. With a "recieve later" option, you would be able to buy two cucumbers, but only receive one at the moment. Then you would be able to come back to the store later to get the other cucumber when you needed it. An initiative like this have already been made by the English supermarket chain Tesco. The initiative is called "Buy One Get One Free Later", and is an initiative for fresh groceries like fruit and vegetables.

To give the consumer better information about when food is going bad is also an initiative that The Danish Ministry of Environment is recommending. By informing consumers of what "Best before", "Last day of sale" and so on means, would get the consumer to use the food more properly. Also by getting the consumer to rely more on smell, taste and looks, rather than just the date of production on the groceries, would lead to lesser food waste.

The Danish Ministry of the Environment also recommends getting discounts even though you don't buy in quantity. Being able to share an offer with another person would let the supermarkets get volumes bought, and the consumer would be able to only acquire the amount necessary.

Only using the groceries necessary is also a way to achieve lesser food waste. Following recipes will help with this if the recipe is specific. If for example the recipe is for 4 persons, and 4

persons will be eating, if the recipe is correct food will not be wasted, and even if there are leftovers after the meal, using these as lunch for the next day, or eating the leftovers the day after, will achieve a lesser waste of food[**madSpild_MindreMadspild**].

Keeping track of products that is being thrown out is another way to reduce food waste. If a pattern occurs you may be overbuying food. An example is if you throw out 500 grams of beef each week, you may be overbuying beef, and might as well buy 500 grams of beef less each week[**madSpild_Greatist**].

Eating the food which is closest to the expiration date is a good way to prevent food waste as well. If the refrigerator is being filled up as the week goes by, new products might be put in the front as they are bought last. By keeping track of the expiration dates by either organizing the products and having the oldest in the front of the refrigerator, or by writing down the expiration date of the different products, could help solve the food waste problem, as the products closest to the expiration date would be used first.²

According to United States Environmental Protection Agency[**madSpild_EPA**], being creative with leftovers is another way of preventing food waste. For example if a loaf of bread has gone stale, cutting it up and using it as croûtons is a great way to reuse it, instead of just throwing it away.

In conclusion; wasting food is a something that we all do, and it is therefore essential to find solutions to minimize food waste. In this section, suggestions have been made, that could help preventing food waste, and therefore help preserving the environment as well. The suggestions have ranged from providing consumers with better information, to providing ideas for supermarkets to help prevent the food waste, that consumers have.

1.4 Foodplan study

Stop Spild Af Mad[**madSpild_RapportAdfaerd**] has conducted a study where eight informants had to create and follow a weekly food-schedule. Each participant had to create a foodplan, in which they could freely choose the different recipes with inspiration from cookbooks and/or the internet.

Planning

The participants had some trouble with pulling themselves together and getting started creating the foodplan, but they were happy to have it, once it was done. It could be beneficial for the person that were in charge of grocery-shopping to be the one making the schedule, since the involvement of other members from the family could increase the time of making it.

Modules

It was almost impossible for the informants to follow the schedule for a whole week without having to reorganize or modify it. Most of the informants made small modules that could be

²Fixme Note: possibly to repetitive

moved around, depending on the family's situation. It was optimal with modules spanning two to three days. Varied meals was preferred, and not simply reheated meals.

Grocery shopping

Seven out of the eight participants were shopping groceries almost every day. This was not seen as a nuisance, but after having planned grocery-shopping for a whole week, they experienced a big relief in their everyday life. This was expressed in terms of more personal freedom as a perk of saving time and money. Planning ahead also decreased buying on impulse.³

1.4.1 Reflection

The participant had trouble getting started creating a new food plan for the week. They also wanted a more realistic food-schedule instead of having to create it from cookbooks. Realistic in the sense, that the plan should fit into their context. A solution to this problem might be solved by automating a lot of the tasks involved with having to come up with the different recipes and do the grocery shopping. By having an application come up with suggested meals based on different aspects such as the:

- Food you already have
- Food you like
- Price
- Stock of stores nearby

People could quickly create the schedule or maybe even have the application create it for them.

1.5 Interviews

Interviews have been conducted, on 5 persons in different households, and the answers can be seen in appendix A.0.1. In this section similarities and differences are looked upon.

1.5.1 General

this subsection contain some of the general questions.

Sex

Two participants were male, and 3 were female, which makes both sexes represented.

Age

³Fixme Note: textbf might be a bad choice.. CHS tried using emphasize on Less gro... and subsubsec on Modules (subsubsec throws a warning because we have no subsec beforehand .. please evaluate, on which type we should use.

The two males are the youngest, at 25 and 28 years, the females are 53, 55 and 75. This gives a broadly spread range, representing a wide target group.

Household

Most households are similar in size, with three households of 2, one household of 3, the last household with 11 persons. This gives a generally good idea of problems that can be found in smaller households.

1.5.2 Food waste

The participants see food waste has the same opinion as the FDB participants to food being thrown out, which is that edible food being thrown out is food waste. Another aspect that was answered by the interviewees is the question of using leftovers, which is discussed underneath. Some participants saw food fed to animals, as not being food waste, and lastly a reason is given for the bundles, in which food is sold, are to big, so the household do not have the time to use it all before it goes bad. The participants were asked if they found if food waste was a problem in their household, and only one saw it as an actual problem, it should be noted that two participants fed food, that was about to go bad, to animals, such as hens or dogs. furthermore if food was thrown out it would mostly be vegetables, and sometimes bread. The participants were asked what they focused on, when they had to decide if food should be thrown out, and all answered that they did not take the best before date into account, but instead looked at the freshness of the food.

Leftovers

All participants say they use left overs, even though the given reasons are different.

1.5.3 Importance

The participants were asked what was the most important about their diet. And the order of their answers varied a lot, but some of the aspects they all had in focus, were quality of the food, and that the prize is good. Other aspects were organic, exiting, healthy and varied meals.

1.5.4 Planning

The participants have in common that they do not plan more then a few days ahead. But they plan their shopping in different ways, one shops for dinner almost every day, another make plans from meat that can be found in the freezer, another is willing to drive far for proper meat, the next wants to shop a few days before the planed meal, and the last person just plan and shop when he arrives at the store.

Shopping

It was hard for some the participants to say, for how long they shop, but two of them shop in 5-10 minutes. Another participant elaborated that she often goes to multiple shops, and therefor the shopping often takes about an hour, in total. Furthermore the participants could not all elaborate when, in the day, they shop, but some said that it would mostly be after work. Next, the

participants shop between 3 to 5 times in a week. Only one of them prefer to plan the shopping in advance, by looking through magazines, to find the best offers. One elaborates, that if he knows what he wants before he goes shopping, it will take a lot less time.

Shopping situation

The participants are affected a little differently by impulsive shopping, but in general, do they like to buy things that are on sale. And some likes to buy the food that is on sale for being close to the best before date.

2. System Choice

In this chapter the three sub activities *Situation, Ideas and System definition* when defining the system, will be described. To describe the system choice we will be using systems development where system choice will be described this way situation, ideas and systems.¹

2.1 Situation

Figure 2.1 shows a Rich picture. This rich picture gives an overview of the users situation from the viewers position. The symbols represent different units, processes and problems. To begin with there will be looked upon the symbols and they will be explained. Afterwards the situation will be explained in depth.

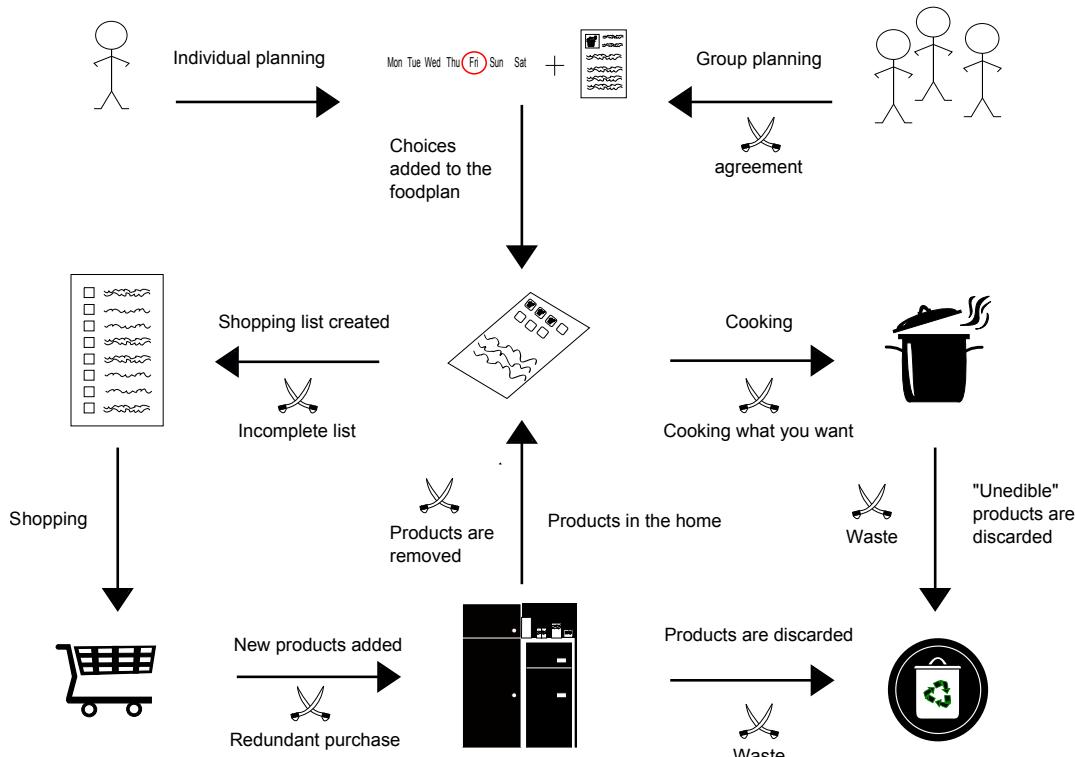


Figure 2.1: A rich picture of the user's situation from the viewers position

² Units includes people, physical objects, organizations and roles.

- On the top line the stick man symbol is used to represent a user and to the top right there is the group of users symbol.

¹ FIXme Note: Brug slides fra lektion 3 SU for at skrive introer til afsnit.

² FIXme Note: resize elementer så de er lige store + kosmetiske fejl

- On the middle line is the overview of the food plan symbol and shopping list symbol which is used to depict the users overview of the two units.
- On the bottom line is the inventory symbol used to depict the household's storage of food products and the trashcan symbol.

Processes includes different categories such as work and production, planning and control and information treatment. In Figure 2.1 all of the arrows between the symbols signifies different processes.

- On the top line there are two arrows pointing into the schedule meal symbol signifying the process of choosing a specific day and a associated recipe. The choice is then added to the food plan showed by the vertical arrow pointing down to the overview of the food plan symbol.
- On the middle line there are two horizontal arrows. The arrow pointing to the left shows how a shopping list is created using data from the food plan. The arrow pointing to the right signifies the cooking work process. Below the middle line there are three vertical arrows. The arrow on the left pointing downwards shows the shopping process, the middle arrow pointing upwards signifies that the food plan is affected by what products that are in storage and the right arrow pointing downwards shows that products can be discarded when the user is cooking.
- On the bottom line there are two arrows pointing to the right. The left arrow signifies that products are added to the inventory after they are bought. The right arrow signifies that products can be discarded from the inventory.

Problems describes the conflicts, contradictions and discrepancies in the situation.

- On the top line there is a conflict under the group planning process. This conflict is named synchronizing and happens when multiple users are scheduling meals.
- On the middle line there are two conflicts. The left one is named incomplete list and happens if the user does not have a complete list over the items that are needed in the food plan. The right conflict is named cooking what you want and describes that users can cook meals that are not on their food plan. Next there are the products removed conflict which happens when a user physically removes a product from the inventory. Next to this conflict we have the waste conflict which happens when food is thrown out during cooking.
- On the bottom line there are two more conflicts. The left one named redundant purchases, happens when the user buys larger quantities of food than the shopping list advises. The other conflict is called waste and happens when food from the inventory is thrown out just as the other waste conflict described just before.

Situations explained

The starting points in Figure 2.1 are on the top line where a user or a group of users plan a meal by choosing a day and a recipe. In this process a conflict can occur when multiple users have to agree on a certain day and a specific meal. When a meal is planned it gets added to the food plan which can contain one or more meals. It is now possible for the user to get an overview of their meal plan and take actions to follow the plan. The first action might be writing a shopping list of what they need. In this process a conflict can occur if the user is not certain on which food products and how much that is needed, resulting in an incomplete shopping list. The user will now take their shopping list and go shopping for the specified items.

The food products are then added to the inventory. During this process a conflict can occur if the user buys larger quantities of products than what was written in the shopping list. The products in the users inventory has an effect on the food plan. This effect is seen when the shopping list is written because products that are in the home should not be on the list. A conflict can occur if needed products are removed and not added to the shopping list.

Another process that starts from the overview of the food plan is cooking. This process can create a conflict if the user cooks a meal that is not planned. This results in products that have been bought for the planned meal is not used and might not get used before they are thrown out. Food products can be thrown out from the inventory and from the cooking process. This can result in food waste if the discarded products could have been used in a future meal.³

2.2 Ideas

2.2.1 Exemplars

In this section, a current IT-system will be described and what other systems that can be used to plan what food to eat. Ideas from other existing systems will be found and put into the context of this projects system.⁴s

Food Planner

One technology that are currently being used to make a food plan is a mobile application called *Food Planner*. In this application the users can plan meals ahead of time, lookup recipes, look at what groceries that needs to be bought, list what the user have in the fridge and appends bought items to the fridge. The following section will take a look at the application, in order to find ideas which will be good in the context of this projects system.

³Fixme Note: Figure 2.1 can also be divided into different context areas. The areas represent contexts in which different symbols and processes are performed most often. The contexts are in the home and out of the home.

⁴Fixme Note: CHS: Still don't know how or if, to incorporate metaphors

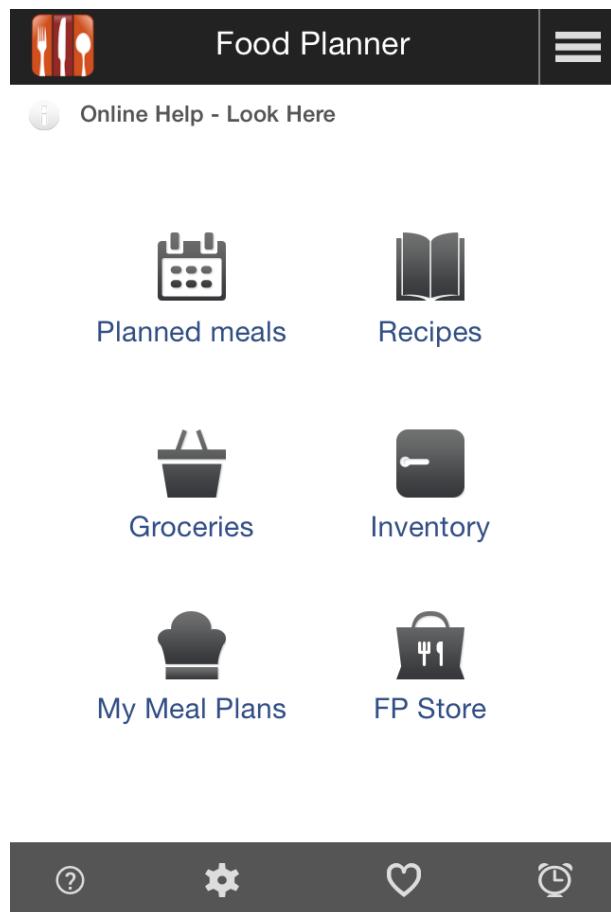


Figure 2.2: An image displaying the index of the application Food Planner

Useful ideas:

- **Inventory:** Each product can be added separately, and the application can be used to keep track on the amount
- **Recipes:** Custom recipes can be created through the application, or they can be found/bought in the application store, or the internet, either by a Google search, or following a link directly to known websites, with recipes.
- **My meal plans:** This menu can keep track of meal plans, that has been added to the application.
- **Planned meals:** In this menu the upcoming days can be planned, with multiple recipes each day, either by adding them manually, or by adding a meal plan from *My meal plans*. Missing groceries for a desired number of days ahead, can then automatically be added to a *Grocery menu*.
- **Registering:** By registering with an e-mail and a password, the application allows the user to back up the information, synchronizing with other devices and sharing is also enabled.
- **Grocery menu:** Here, groceries that needs to be bought, can be added either automatically from *My meal plan* or added manually. If an item is not known to the program, it can be added using a bar-code scanner, and naming the item manually.

Ideas used in the context of a new program

- **Inventory:** To help avoiding food waste, this idea would be useful. By prioritizing the inventory, the food in the home, has a better chance of being used, before it expires.
- **Recipes:** Using Recipes, people can try new meals, and if the ingredients list, needed for the meal is known, the Inventory can be taken into account, or a list of needed ingredients can be added to the grocery menu.
- **My meal plans:** With this idea, it is easy to save different diets, and lists of favourite meals, etc.
- **Planned meals:** Knowing what food is planned ahead, new recipes can be added to reduce food waste and/or variate the meals.
- **Registering:** This idea enables the use of synchronizing, which could be useful if the program is used in a household of more than one, or if it is to be used on multiple devices.
- **Grocery menu:** This menu is very useful, for the use of a shopping list, where it can be tracked what has been bought and what still needs to be bought.

Website

Another alternative could be a tool found on the website <http://realfood.tesco.com/meal-planner.html>^[tesco_food]. The website has six main menus; About meal planner⁶, Create your own, Leftover inspiration, Featured food plans, Customer food plans and My binder, for now these menus will not be explained, but they have been looked through, to find useful ideas, before its looked upon, how they can be used in the context of a new program.

Useful ideas:

- **Meals:** When creating a food plan, the user can chose to plan breakfast, lunch and/or dinner.
- **Surprise me!:** With this feature, an food plan is automatically created, which is a fast way of planning, if you are not afraid of what you might have to eat.
- **Cooking time:** The user can also define if the cooking time must be 30 minutes or less, 60 minutes or less or if the user does not mind.
- **Dietary preferences:** When creating the plan, the user can also set preferences such as; vegetarian, egg free, gluten free and low fat, furthermore the user can add liked and disliked ingredients, before searching for recipes. Another preference, which can be set here is if the user is on a budget.
- **Leftover inspiration:** The website offers features to lower food waste, by having articles on how to use leftovers and cook meals so they are easy to save for another day. Lastly they have an option to list leftovers and then recommend recipes from this list.

⁵Fixme Note: "Should I add more pics of Food Planner app?" Christian asked the group.

⁶What we call *food plan*, the website calls meal plan. This report will from here on, use food plan where the website uses meal plan.

- **Featured plans:** It is possible to browse the websites featured food plans, either for inspiration, or to add the entire food plan, to your own.
- **Customer plans:** In this menu the users, or customers, of the website can add their own food plans or browse other users food plans, when browsing, it is possible to filter by; vegetarian, quick and easy, family, dietary preferences, healthy and budget.
- **My binder:** With this menu it is possible to bind recipes and give them notes. The website then offers a feature to search and filter the saved recipes by ingredient, cuisine, dietary requirements and cooking time, if the meal has been cooked yet, or a simple alphabetic order.

Ideas used in the context of a new program

- **Meals:** If a user has a company deal including lunch, it will not be necessary to plan lunch, therefore this idea could be optimal to include.
- **Surprise me!:** Is a good function for the user who wants new inspiration and cannot decide on what to have.
- **Cooking time:** Some days are more busy than others, therefore adding this feature would be ideal.
- **Dietary preferences:** If a user would like to follow a specific diet, it would be ideal to let the program help with finding recipes for the specific diet.
- **Leftover inspiration:** This function is ideal to help avoid food waste.
- **Featured plans:** Not having to create all meals on your own or gives new inspiration.
- **Customer plans:** This feature allows for recipes with personal twists, and allow a user to follow custom plans.
- **My binder:** This is an easy way of organizing interesting recipes and food plans.

2.3 System

2.3.1 Definition (FACTOR)

To formulate a system definition we used the FACTOR [OOAD_BATOF] criteria to establish important parts of the system.

Functionality Help users organize a meal plan based on their preferences and inventory. Create/edit user, inventory and plan.

Application domain Database responsible employees⁷, Users planning meals.

Conditions The system will be used by users with varying technical abilities and cooking experience.

Technology Tablet, smartphone

Objects Family member (user), inventory (groceries), recipe

⁷FIXME Note: what is this (explain in definition below)

Responsibility An administrative/planning tool

2.3.2 System Definition

A system which primarily lets you plan your meals in advance based on what is in your inventory to assist the user and reduce food waste. Secondarily the program will allow administration of a meal plan and inventory and incorporate functions to alter this plan and inventory by adding or removing recipes and groceries. The system will be designed for tablets and smartphones, as these devices offers mobility and thereby available both in and out of the user's home.

3. Problem Domain

In this chapter the reality of which the user is going to see will be described. This reality is constructed of the areas which the system is going to administrate, monitor or control. Descriptions of classes, objects, structures and behaviours will be made throughout the chapter.

3.1 Classes and Events

The purpose of this chapter is to select and analyse components from the problem domain. These components are split into classes and events.

3.1.1 Classes

In this section a list of classes will be presented. The classes **there is described** are those which have been chosen through a class candidate analysis.

Physical

- Product

This class is used to identify different food products for the recipes with information such as food type and expiration date. This class is essential for the program and will be used frequently.

- Recipe

This class holds information about the food ingredients and instructions on how to cook the meal.

- Scheduled Meal

This class contains information about a recipe and when it is scheduled to be cooked.

Persons

- User

This class contains information about the preferences of a specific user, and will allow the solution to synchronize/share data with other users or devices.

Places

- Shop

This class holds information about the location of specific groceries and sales.

Other

- Achievement

Information about point, progress and criteria.

3.1.2 Events

In this section a list of events will be presented. The events there is described are those which have been chosen trough an event candidate analysis.

Consumption



- Budget used

Trigger: The balance have been changed as products have been bought.

- Budget refilled

Trigger: The user sets a new weekly/monthly balance, or reaches a new time period.

- Product bought

Trigger: A shopping list have been completed. The items bought will be added to an inventory.

- Product removed

Trigger: A meal have been prepared and the product should be removed form the inventory, or have an subtraction from its original volume/quantity.

- Product expired

Trigger: When a product reaches its expiration date it should be thrashed" and removed from the inventory.

Planning

- Recipe scheduled

Trigger: When a recipe is tied to a date in the meal plan.

- Recipe removed

Trigger: When the user removes a recipe which was tied to a date in the meal plan.

- Shopping list item added

Trigger: When a recipe have been scheduled in the meal plan, the system will check for which food ingredients is needed, and thereafter adds any missing products to the shopping list.

	Product	Recipe	User	Meal	
Consumption					
Product added	+				
Product removed	+				
Product expired	*				
Product unexpired	*				
Product quantity changed	*				
Product expiration changed	*				
Planning					
Shopping list item added	+				
Shopping list completed	+				
Meal added		+		+	
Meal removed		+		+	
Meal rescheduled		+		+	
Meal participants changed					*
Meal Date changed					*
Day passed					*
Other					
Preference changed			*		
Recipe found		*			

Table 3.1: An event table for the program

- Shopping list created

Trigger: When a meal plan is completed, by which there are no days without a recipe, the shopping list will be locked and considered completed as well.

Preferences / settings

- Preference changed

Trigger: When the user goes to a settings menu to ban products from the system, due to allergies, diets or preference.

Other

- Achievement unlocked

Trigger: A user has fulfilled the requirements for a achievement.

3.1.3 Event Table

An event table, on table 3.1, have been constructed in order to get an overview of the relations between classes and events. The event table allows for a better judgement of which classes are relevant in the program.

3.2 Structure

In this section we will be describing the overall structure of the program, through class diagrams.

3.2.1 Individual class description

In this subsection the different classes will be described with a short text.

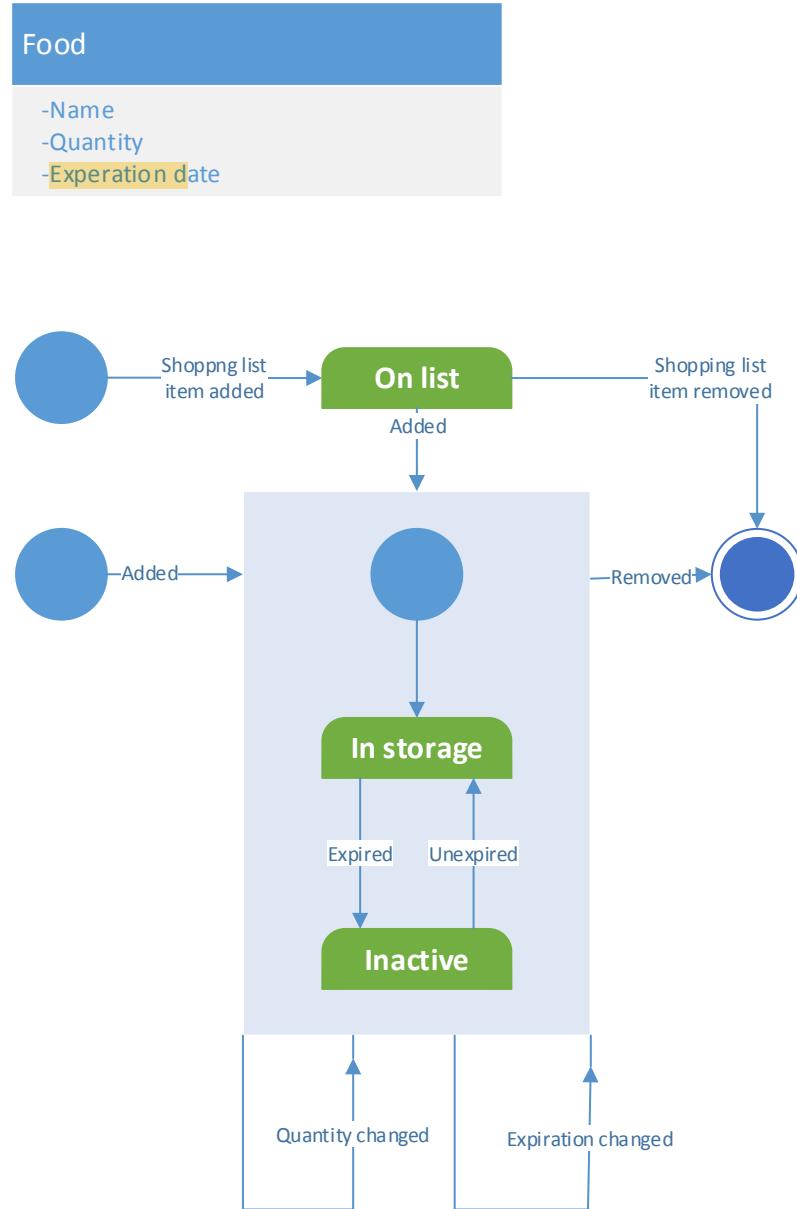


Figure 3.1: Class diagram for the food class

The **Food** class contains information about the quantity and expiration date of a food item. The food items can be found in the lists inventory or shopping list.

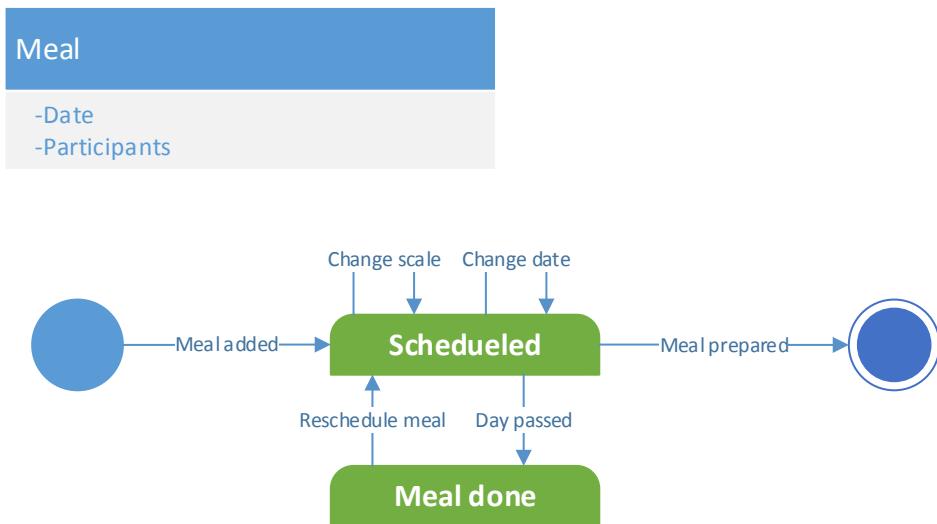


Figure 3.2: Class diagram for the meal class

The **Meal** class contains information about a meal and when to make it. It consists of recipes.

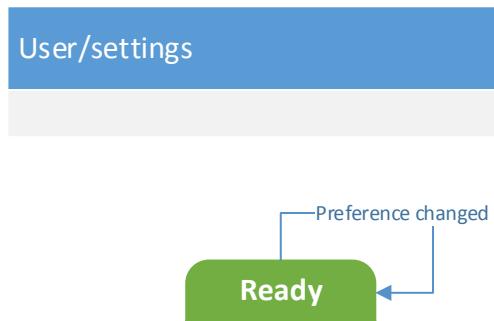


Figure 3.3: Class diagram for the user settings class

The **User settings** class contains preferences from the user which the system should take into consideration.

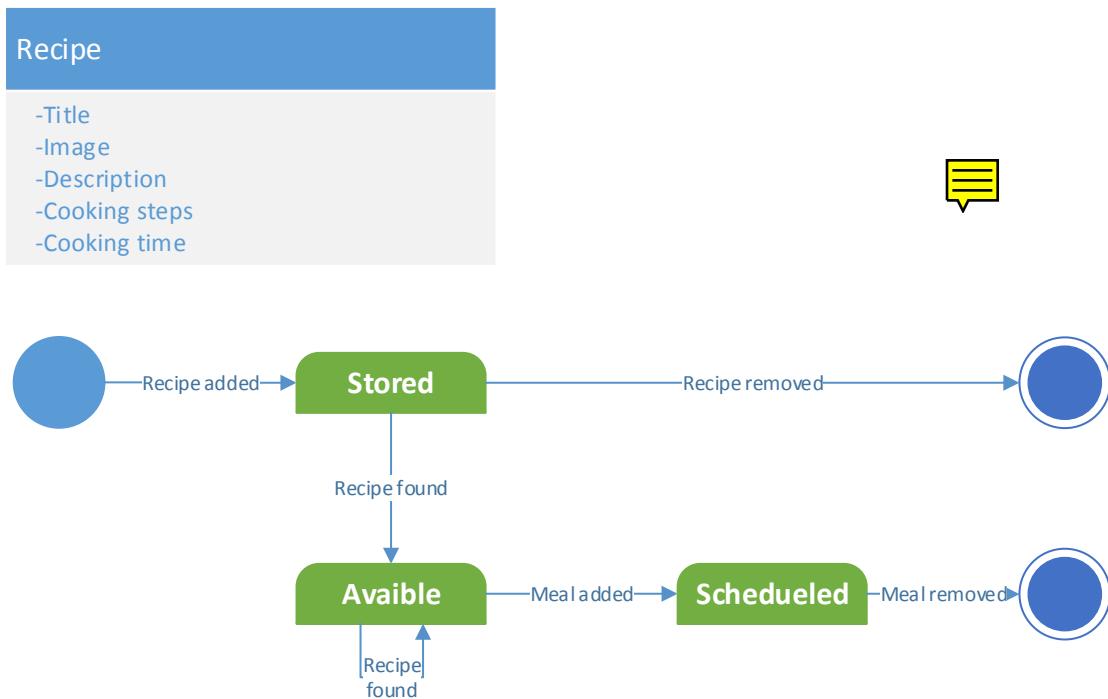


Figure 3.4: Class diagram for the recipe class

Recipes consists of **Food**. It also contains information on how to prepare the meal from the recipe.

3.2.2 Class diagram description

The class diagram is structured with **Food** as the central class. *Shopping List*, *Inventory* and *Foodplan* are lists, containing *Food* or *Meal*.

Food objects can be part of a **Shopping List**, a user **Inventory**, or be part of a **Recipe**, seen as aggregation on Figure 3.5. A **Recipe** contains 1 to many Food objects, and is part of a **Meal** that in turn is part of a **Foodplan**. A **Meal** contains exactly 1 recipe, while the *Foodplan* can have any number of planned *Meals*.

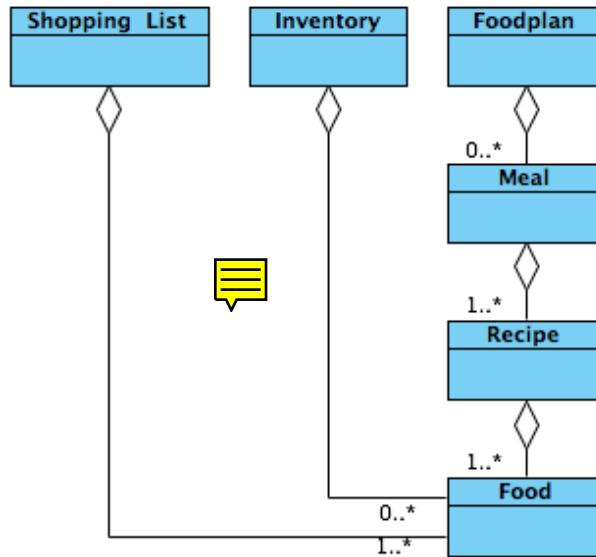


Figure 3.5: Relation between classes.

3.3 Priority of criteria

When designing a software based system, it is important to consider which criteria is needed for the system, and if some of the criteria is more important than others. In this section the criteria is going to be put into a table, in the table the criteria is listed, and the importance of them range from very important to less important. The table also states if the criterion is irrelevant or trivial.

Criteria	Very important	Important	Less important	Irrelevant	Trivially fulfilled
Useful		X			
Secure			X		
Effective			X		
Correct	X				
Reliable			X		
Maintenance		X			
Testable		X			
Flexible		X			
Understandable			X		
Reusable				X	
Movable				X	
Integrable			X		

Useful indicates the adaptation of the organisational, work related, and technical surroundings. This is important for the food planner system to fulfill, since the organisational surroundings indicates that the user has to be able to organise the meals correctly.

Secure indicates if the system is secure against unauthorized accessibility. This is less important in the food planner system, since the worst security breach would just reveal what people eat.

Effective indicates the economical usage of the technical platform facilities. This is marked less

important, as smartphones have different technical facilities, and therefore they can not all be used. Furthermore it is not needed to use all these facilities for the food planner system.

Correct indicates the level of fulfillment of the formulated requirements. The correctness of the food planner system is very important, as it is needed to live up to requirements found in the analysis part for the system to be effective.

Reliable indicates the fulfillment of the requires precision when functions are executed. The reliability is less important for the food planner system, as functions can work a little different than stated, as long as the user gets a usable result.

Maintenance indicates the cost of finding and correcting errors in the system. The maintenance of the food planner system is important, as error will be made throughout the project. Therefore it is important that error can be fixed for the system to work properly.

Testable indicates the cost of securing that the system fulfills the requirements. Is important for the food planner system to be testable, as testing is part of the waterfall method, and also a good way for the system to be maintainable which is important to the system.

Flexible indicates the cost of changing the system once it has been taken into use. Flexibility is important for the food planner system, as changing and updating is going to be done after the release, either because new functionality is needed, or just to load in more recipes in the program.

Understandable indicates the effort to make sure that the system can be understood in the context. Understandability is less important to the food planner system, as the system will not be released for other developers to work on.

Reusable indicates the possibility to reuse parts of the system in similar systems. Reusability is irrelevant as the system is not going to be released, therefore it will not be used in similar systems.

Movable indicates the cost of moving the system to another technical platform. For the food planner system it is irrelevant to consider the movability, as the system is not going to be moved to other technical platforms than smartphones and tablets.

Integrable indicates the cost of connecting the system to other systems. Integrability is less important for the food planner system, because the system probably will not be connected to other systems, unless API's is going to be included.

4. Application Domain

4.1 Usage

This chapter will describe the systems interaction with its surroundings. This will be done by making an actor specification and then presenting user pattern diagrams, which will be showcased in state machines.

4.1.1 Actor specification

An actor will be described by an actor specification , the actor is a user of the system.

User

Objective: A person who prepare their meals themself, either for themself or for a household. The user's primary need is to plan their week in order to efficiently shop their groceries and lessen food waste.

Characteristic: The system includes a user base of which the users have different needs and preferences.

Example one: User A is an allergy sufferer and has to make meals which excludes nuts. When A is shopping she needs to look at the package info to make sure that the product does not contain trace of nuts.

Example two: User B is a young student who is in a relationship. User B has a tight budget and a little time frame to shop in. B finds it difficult to sustain an overview of their share storage of food. Therefore B will sometimes buy products they do not need, for example, B might purchase milk even though they already have three litres stored. This will sometime result in them not being able to use all the milk before it expires, which is bad for their shared budget and food waste.

4.1.2 User pattern

The user pattern shows an abstraction between the interactions in the system and actor. State machine diagrams are used to show these interactions. The diagrams show how the dynamic states can shift trough interactions with the actor. Even though many of the details is excluded, it still gives a good overview of how the logic is set up in the user pattern and how the the dynamic flow goes.

The purpose of the diagram is to create an overview of the application domain's interactions with the system. This will be used to find acquirements for the functions and user interface. There are three user pattern diagrams, each representing different areas of the system. The three diagrams showcase the user patterns in the:

- Shopping list
- Foodplan

- Inventory

When the user actor traverses through the system, the actor will undertake one of two roles. An administrative or a planning oriented role. The option to go back or cancel have not been drawn on the diagrams as their only contribution is to make the diagrams less readable and add redundant repetitions, that goes for all of the diagrams.

This diagram is for the foodplan, in this part of the system the user actor operates under the planning role.

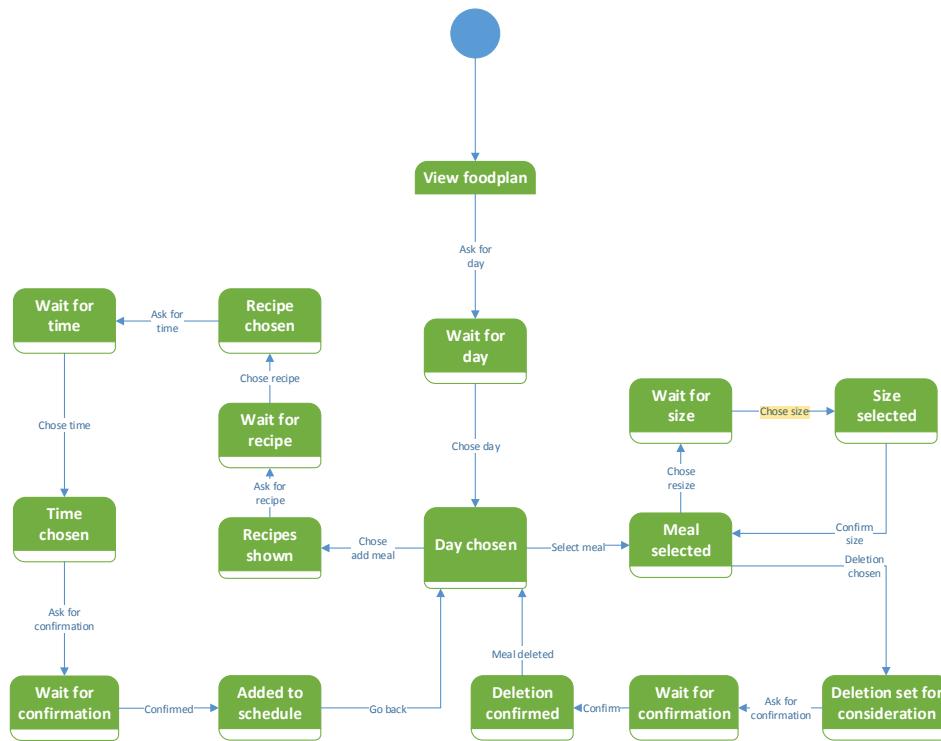
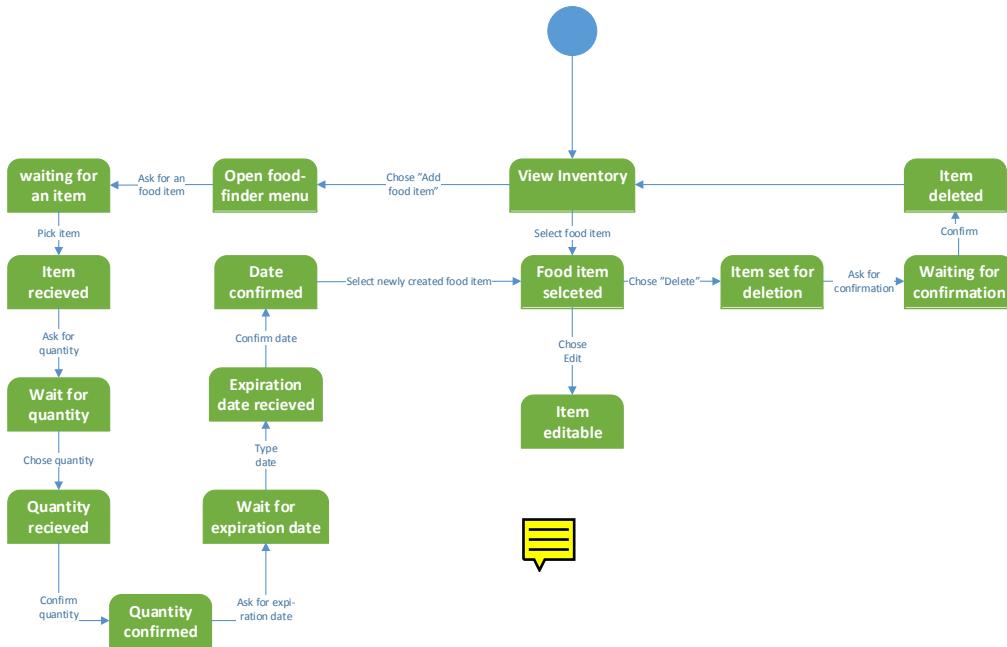


Figure 4.1: The view foodplan state diagram

When the user goes to the foodplan part of the system, he will have to choose a day to view the foodplan for. From there the user can add a new meal to the plan or select an existing planned meal. If the user chooses to add a new meal, the system will ask for a recipe. The recipes the user can choose from varies depending on the user's preferences. If the user have chosen to avoid some products, recipes that contain those products will be excluded. When a recipe have been chosen, a time for when on the day the meal should be prepared for will have to be entered. After the time has been entered, the meal will be added to the schedule.

The next two diagrams is for the inventory. In this part of the system the actor operates as a administrator.

Figure 4.2: The *view inventory* state diagram

The user starts out on the "view inventory" state where the inventory is shown. From here the user can choose from three different options: One that adds food to the inventory, another which allows the user to delete existing food items and a last option for editing food items. The edit option has its own diagram (see fig. 4.3). If the user chooses to add food, they will have to select food from a food list. When a food item have been chosen, the system will ask for the user to select an quantity in order to know how much of the food item the user have acquired. For example, the user can buy five bananas or two hundred grams of flour. The user will afterwards have to enter when the food expires. When this is done the food will be added to the inventory. This leaves the user at the food item they have added, as if it were selected on the inventory screen. This is the same result as if the user chose to select an food item from the beginning. From here the user will be able to edit or delete an food item.

When the user chooses to delete a selected item the system will ask for an confirmation for that user wanted to delete. If the user confirms, the system will delete the food item and go back to the view inventory state.

The following diagram, is the last diagram which also covers the inventory. In this part of the system the actor operates as a administrator.

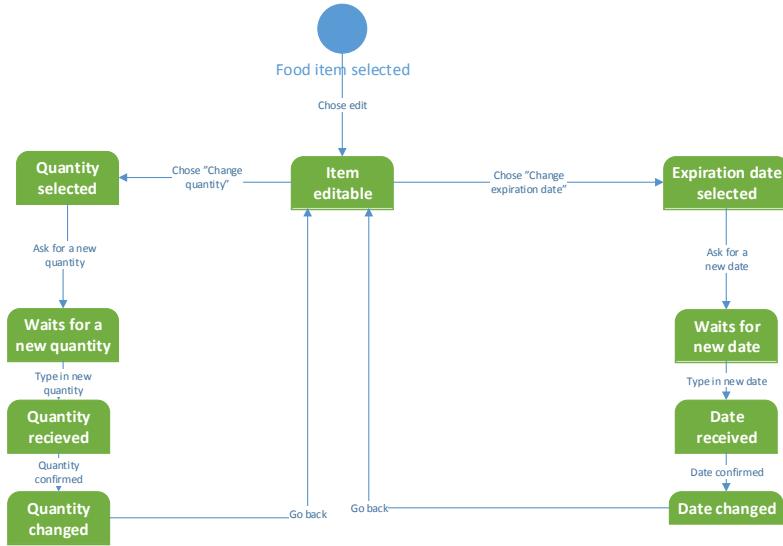


Figure 4.3: The edit inventory state diagram

The possible options in form of editing is to change the quantity or expiration date. When choosing *Change quantity* it works like the *Change quantity* in the shopping list, which means that choosing an amount that leaves it at zero deletes the item, and leaving it below zero result in an error. When the user tries to change the expiration date, the system prompts for a new date. All dates will be valid, which means that an item which was expired can go "unexpired" as described at the diagram fig. 3.2. This also works the other way around as an fresh item can expire if the date changes to the day prior to the day it expires.

Summary

The actor specification helped to establish a understanding of how the user base will interact with the system. The user patterns helped to reinforce that understanding. By making the user patterns, some flaws in the design were discovered which resulted in varies tweaks and reworks.

4.2 Function definitions

The diagrams in section 4.1 contain different functions these functions have been given a name and listed in the following tables, where they have been assigned a complexity of Simple, Medium or Complex, and they have been assigned one or more of the following types; Read, Update, Calculate or Signaling. The more complex functions are further elaborated under each table.

Table 4.1: Shopping list

Function	Complexity	Type
Update shopping list	Complex	Update
View shopping list	Simple	Read



Update shopping list: The update function will be triggered by the signal function from the inventory "Update shopping list" function. When a meal is added or removed from the food plan. The shopping list should represent what ingredients that is needed to prepare the meals in the foodplan.

Table 4.2: Food plan:

Function	Complexity	Type
Add meal	Complex	Calculate, update
Remove meal	Simple	Update
View food plan	Simple	Read

Add meal: The add meal function is given the complex status as it involves many actions in order to add the meal to the schedule. When the user goes trough the steps necessary to schedule a meal, the system will do a lookup in order to create a list of needed ingredients. This list will be compared with what the user has in the inventory, and any ingredients that is not present on the list will then be added to the shopping list. This is were the previously described function comes into play.

Table 4.3: Inventory:

Function	Complexity	Type
View inventory	Simple	Read
Update shopping list	simple	Signal
Add Item	Complex	Update
Edit Item	Complex	Update
Remove Item	Simple	Update

Add item: The add item function have been marked as complex. The function is complex as there are many steps involved when adding an item, see Section 4.1.2 for a description of these steps. By having this function the user is able to add food to their storage manually.

Edit item: The edit item function is complex as well. If the user changes the expiration date of an item, it could potentially update the inventory as an item can "unexpire". The same goes for changing the quantity of an food item. If the quantity reaches zero the item will be deleted, and therefore an update in the storage will be needed.

5. Problem Statement

It is a problem that so much food is going to waste. This is a consequence of bad planning when people are shopping for their upcoming week, and because the consumers want fresh groceries. Miscalculations are bound to happen if the one doing the shopping is part of a bigger group, such as a family. Shopping is also time consuming if people want their groceries at a low price. It can be hard to balance the price and time if the cheap stores are far away, or if the consumer is unaware of recent discounts. It becomes extra time consuming if the shoppers have special food diets or preferences. To maintain a low foodwaste level can also add to the difficulty of planning the day, but using all of the ingredients, and to extend that keep track of what is in the inventory. Users who live alone also find that they have a big foodwaste, and have a higher cost per meal than users who live and eat together with others.

The problems that was found can be summed up to the list:

- Too much food is being wasted.
- Miscalculations happen when more people live and eat together.
- Shopping is time consuming.
- Diets is hard to take into consideration.
- Consumers living alone have the biggest foodwaste per person.
- Consumers living alone have a higher cost per meal, than people living with others.

How is it possible maintain a low foodwaste level, and still want to be able to fulfill their food lifestyle choices. This can be summed up in the following list:

- How do you keep a low food waste level?
- How can a solution fulfill their food lifestyle choices?

Part III

Design

6. Scenario-based design

In this chapter the design method, seen on fig. 6.1 is followed to create the interaction design for the program. The method focuses on the use of scenarios, which are about people performing activities in context by using technology, throughout the design process. In the model there are different design processes shown as clouds and the products of these processes shown as the square boxes. Each of the main processes; Understanding, Envisionment, Evaluation and Design, in *designing interactive systems* course are used in the scenario based method. The model, fig. 6.1, is rather complex and has therefore, in this report, been split into three sections, Model Development in 6.1, Prototype Development in 6.2 and Design Development in 6.3. Model Development uses the product *User stories* and *Conceptual model*, which are processed with the *Understanding Process*, to create the products; *Requirements* and *Scenario corpus*, which then are used to create the final product of this section, a *Conceptual model*. In the second section, Prototype Development, a new product, *Concrete scenarios* is created from *Conceptual scenarios*, *Concrete scenarios* are then processed to create paper prototypes. Lastly the third section, Design Development, combines the two previous sections to define a *Design language*.

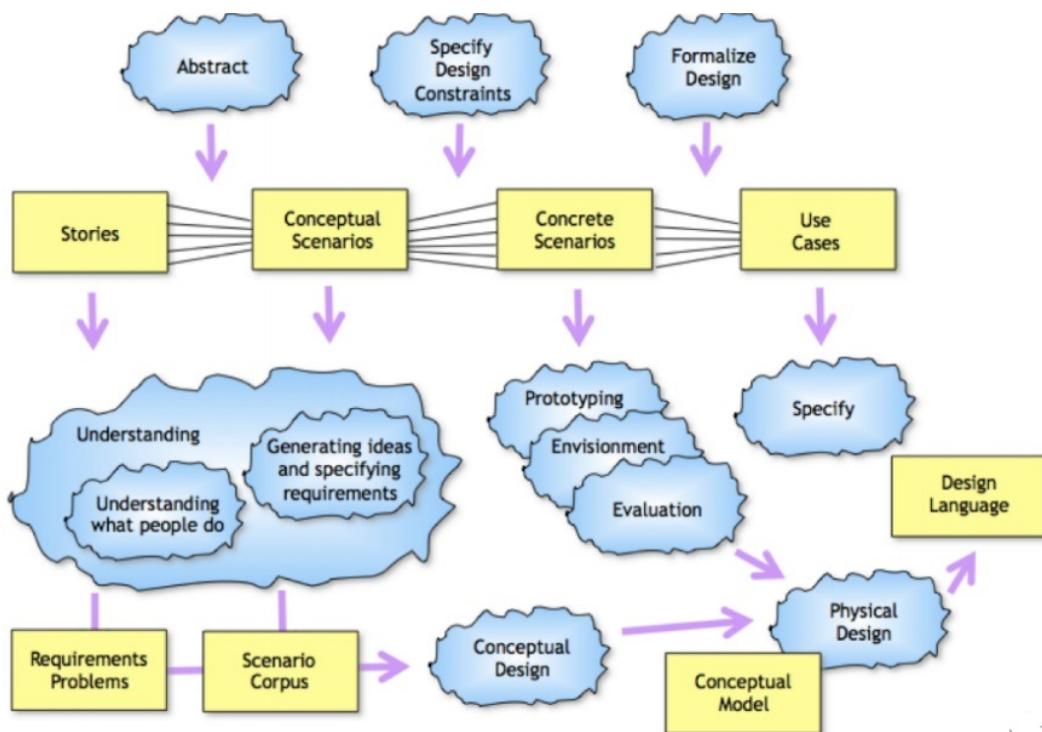


Figure 6.1: Scenario-based design method

6.1 Model Development

As explained in the introduction of this chapter, this section will take a number of user stories, which are used to define conceptual scenarios. From these an understanding process includes personas, which can be found in appendix C, the personas are created for the purpose of this process, which is used to create two products; a list of requirements and a scenario corpus. The list of requirements are briefly explained in section 6.1.3, and the scenario corpus also includes

this requirements list, to define a set of abstract scenarios which have been carefully thought through.

6.1.1 User Stories

All products and processes of the design method springs from the product *user stories*, which are shown below. The stories are all different persons who have described their last shopping situation, the stories vary in detail, but can all be used to find different actions, requirements and problems, which is some of the aspects that are used in the following processes and products of the method.

Person A

I had invited guests for dinner Monday evening we were having kale and ham, both something i had in the freezer, actually i had two hams in the freezer, because i bought one on a sale the day before, and that way i would be sure that there were enough meat, in this new and bigger ham. The only thing missing for the evening meal is now two liters of skimmed milk, so off i went to the grocery store, with my dog in a leash. We walked to the closest store, which also happens to be a big store. When i enter the store, i begin to doubt if the ham really is big enough, so i look in the fridge with sausages, and there happens to be a Christmas sausage, and at the same time on sale, so thinking "better safe than sorry" i put the sausage in my basket, furthermore i think to myself that any spare meat can be used as cold cuts, in the days to come. And now i head for the two liters of milk, and then the checkout. On the way i pass the vegetable area, and see dried figs, who can resist such a thing, so into the basket with those as well. Now i get through the checkout, and past the baker on my way out to the dog, the dog!, i cannot forget to get her something, so i visit the baker to get a sausage roll, but they have a sale, where you can buy 5 rolls, so why should i not do so?

Person B

Last time i went shopping was when we in the household were to eat together, another person in the household looked through the kitchen, to find out what was missing, and then told me; butter, potatoes, carbonated soft drinks, and dessert. I went to the closest store, and bought what was needed. When i returned to the apartment i decided that we should have the drink "white Russian" as dessert, we had all we needed except milk, so off i went again, thinking that this extra trip could have been avoided if we had talked about what dessert actually was meant, before i went for the first trip.

Person C

Knowing that I want to shop the day after this day, I put a piece of paper on the table, and during the day, every time i remember basic stuff that we need, I write it down, things like, paper towels, rice, soap, and so forth. Later during the day I sit down to look through recipes to figure out what to have for dinner the next couple of days. I match recipes with what we already have at home, so that we can clear some of the storage, and in that way save some money, now that

we are close to Christmas, and the budget therefore is a bit more tight, than normal. Because I based the meals on what we had at home, the shopping list did not get that long, therefore i was determined only to have to go to one store, the day after. The next day, the day i had planned for the shopping, did not go as anticipated since both my husband and I had to work until late that day, therefore, I only went to get the most critical items on the shopping list, making an easy dinner for the evening, and groceries for the kids to bring as lunch the following day. During the shopping I remembered that my smallest child had to bring an advent present, to school the next day, so I bought this as well, and the rest of the shopping list had to wait till the day after.

Person D

The last time I had to go shopping, was to get groceries for a Christmas lunch. My boyfriend and I had to get cabbage and cream to make kale, and we had to get marinated herring. Preferably we wanted to get herring marinated in tomatoes, and the cream has to be organic, because I am allergic to milk products that are not organic. Because of these requirements we chose to shop in one of the biggest malls, in the city. But to our big surprise, they did not have any marinated herring other than the ordinary, and they only had 1/4 liter organic cream. Because of this we therefore chose to get in another store. Before we got to the checkout I found a Christmas-gnome costume, for my niece, which I wanted to get for her, and to go with this costume we had to get a pair of matching tights as well. Furthermore my boyfriend found a cheap bathrobe, which he has wished for, for a long time and he therefore chose to buy it. We now went to the next store, to get the marinated herring and the half liter of cream that we are still missing. We found the cream, but we were still looking for the herring and some marinade. We did find some variations of marinade, but not any tomato marinade, which was our desired type. Never the less we chose to go with the another type of marinade, so we did not have to visit anymore stores. On the way to the checkout, we also stumbled upon chocolate ideal for melting, which we were missing a few days before, when we wanted to make some Christmas candy. So we bought the items and went home.

Person E

Last time I went shopping, I wrote to my girlfriend, over the internet, to see what was needed. Since she was at home she could quickly see this, and since I was at school i could easily visit the grocery store on my way home. First of all, we needed to figure out what to make for dinner. After dinner was decided my girlfriend quickly went through the kitchen to see what we needed. After she saw what was missing she texted me a list of missing items for dinner. When I was off school I texted my girlfriend to verify that we only needed the items she had texted me earlier. When I came to the grocery store, I started shopping after what was listed in the text message, and bought all the items, though a few items was sold out, I therefore called home to inform that the needed items was sold out, and what item I needed to purchase instead. After the shopping was done I went home, only to find out that due to that I am the only one of us eating bacon, it was not put on the shopping list, and therefore I had to go out again to get the needed bacon.

Person F

I live quite a long way from any shops and therefor I prefer to only shop once a week. And i prefer to only shop in one specific store. Due to the fact that i live long from the store, I do not return to the store, if I find that I forgot something. During the week I note on a list what I need to get, so I do not have to walk around and remember it all the time. Before I note the items on the list I check to see if we have it in the freezer, garden or in the storage. So I do not get to much of something. Furthermore I look through recipes if I have to cook or bake something that I am not used to make.

Person G

I rarely write down what I need to get, and therefore I mostly have to go shopping every day. Often I look through my inventory when I have to go shopping for something, to see if there is something else I am missing. Furthermore I normally plan what to buy when I know what kind of sales there are at the time. And if it is really good sales, I do not mind going to more than one shop. Lastly i often get so much from these sales, that i have to put some of it in the freezer, and in that way i have food for other days.

6.1.2 Conceptual scenarios

In this section conceptual scenarios are defined through an abstract view on the user stories in section 6.1.1. The scenarios consists of generalizations of situations that are alike, and will together with the user stories be used in the following understanding process.

Conceptual scenario 1

People who plan to make a meal look in their storage, freezer and cabins, to see if meals can be created from what is already at home. If the shopper have not created a shopping list, she/he can get confused about what really is needed, thereby buying to large quantities, furthermore they easily get tempted to impulsive shopping. Lastly, if there is not created a shopping list, shopping will have to be done more times during a week, because items can be forgotten and because it is harder to plan a longer period in advance, when you have to remember all you must buy.

Conceptual scenario 2

If more than one person is making a shopping list, and this shopping list is not detailed enough, it can result in getting wrong items or missing to buy certain items. By having this shopping list to follow, users are less tempted to do impulsive shopping.

Conceptual scenario 3

By planning long time in advance, a shopping list can be created over a longer period of time, where items are added, when the person who is writing the shopping list, remembers what is

needed. This also allows for good time to go through recipes and compare these to the inventory of the household, this minimizes the need for items to bought, and thereby helps to minimize food waste. Lastly, as with scenario B, having a shopping list reduces the amount of impulsive shopping.

Conceptual scenario 4

When shopping in a unfamiliar store, the shopper will be less certain of what actually can be bought in the given store, therefore the likelihood of having to go to multiple stores are greater.

6.1.3 Design Requirements

When analysing the interviews in section 1.5, conceptual scenarios in section 6.1.2, the PACT analysis in appendix B.0.2 and the personas appendix C, some design requirements are discovered. These requirements have been listed in the appendix, appendix D and are later included in the definition of a scenario corpus. Furthermore the long list of requirements have been sorted into the same domains as can be found in scenario corpus, section 6.1.4, but also includes a list of more general requirements followed by a list to requirement actions, as well as a limited list of technologies, which can be used. Some Requirements can be placed in more than one domain, but is only be listed in the domain that seems most ideal.

6.1.4 Scenario Corpus

In this section a scenario corpus is described. The scenario corpus will be based on the previous part of this section, as well as the interviews section 1.5, the PACT analysis appendix B.0.2, the personas appendix C, and lastly the design requirements section 6.1.3. To make the scenario corpus, similarities and differences of these parts of the report are found, each of the three scenarios will cover a domain of requirements, and some general requirements as well. The first scenario mainly include planning of meals, the second scenario mainly include shopping, and the last scenario mainly include handling inventory. Other domains to find are recipes and a general domain.

The first scenario

Jane lives with her boyfriend Jacob, and is the one to normally do the planning, it is Sunday, and she decides to plan the meals for the next weak, while Jacob is out for football training. Furthermore have they just started a new diet, where they try to avoid certain foods.

1. Jane sits down to start the application to help her plan the next week of meals. Starting from today she looks at recipes and sees there is a recipe when they only need to get carrots. Jane texts her boyfriend to tell him to buy some carrots for tonight's meal, she then contentious the planning
2. Thinking that carrots are normally sold in big bags, Jane schedules four of the meals in the coming week, to have recipes where there are used a lot of carrot.

3. Jane plans the last three meals to be placed in between the four meals with carrot, so they would not have to eat too much carrot.
4. Jane and Jacob are having some friends over the next Saturday, therefor Jane also plans this meal for five persons instead of the two, that they normally are.
5. Later Jacob comes home, and he remembered to buy carrots. Unfortunately for Jane's planning he only bough a small box of carrots, because he had read in the text message that it was carrots for tonight's meal, and he therefor did not want to end up with too many carrots.

The second scenario

It is now the day after and Jane is going to do some shopping after work, she knows that there are sales in two stores, placed close to each other, so she drives there, to do the shopping

1. When she arrives at the first store she opens her application with the shopping list, she does not want to shop for the entire week, because she knows she will have to buy a lot. So she sets the shopping list to show what needs to be bought for the next three days.
2. She sees that she has to buy a lot of carrots and remembers that she had scheduled four meals with carrots, because she thought her boyfriend would have bought a large amount the day before. But she does not want to have four days with a lot of carrots, if she can avoid it, so she reschedules two of the meals, before she contentious the shopping.
3. Jane walks past the shelves with candy when she sees there is a sale on chocolate, she would like some for the evening, but is also quite certain that they have some at home, so she opens the inventory list, only to see that they barely have any chocolate left, therefore it goes into the basket as well.
4. Jane knows what there is on sale in the other shop, and therefore she does not buy it in this store, also there are some items which she cannot find, she therefore proceeds through the check out and crosses off the bought items, before going to the next store and getting the rest.

The third scenario

Jane is now home from the shopping, and it is time to organize the inventory and begin cooking.

1. Jane gets home and begins to put the bought items away, when she gets to the chocolate, which she bought even though it was not on the shopping list, she remembers that she has to manually add this to their inventory, because it has not been automatically added from the shopping list.
2. After putting away all the items, she begins to cook, Jane has never cooked this meal before, and therefore rely heavily on the recipe. she starts by finding the needed ingredients from a list which she can easily scroll through with one finger.
3. after finding all the ingredients she begins to cook, the recipe requires Jane to use her hands, which makes them greasy, furthermore when she has to check up on the amount she needs of certain items, and when she has to go further into the preparation steps, she has to

scroll, but with her greasy hands she must do this with her elbow, which she can, because the touch area for scrolling is quite large.

6.1.5 Conceptual Model

Figure 6.2 shows a conceptual model between different domains of requirements and actions. This conceptual model is created to give an overview of what has been defined in the rest of Model Development in section 6.1. The four boxes with sharp corners are seen as domains, some of which have been used in the scenario corpus. Recipes has not been used as one of the main domains in scenario corpus, but includes the information need to cook the specific recipes. Another domain that is not seen in fig. 6.2 is the general domain, this domain contain requirements which are not directly influencing or needed in the other domains, which is why it is not included in the model. **the** other boxes with the soft corners are requirements to the different domains, lastly the arrows with text are actions that sets new requirements.

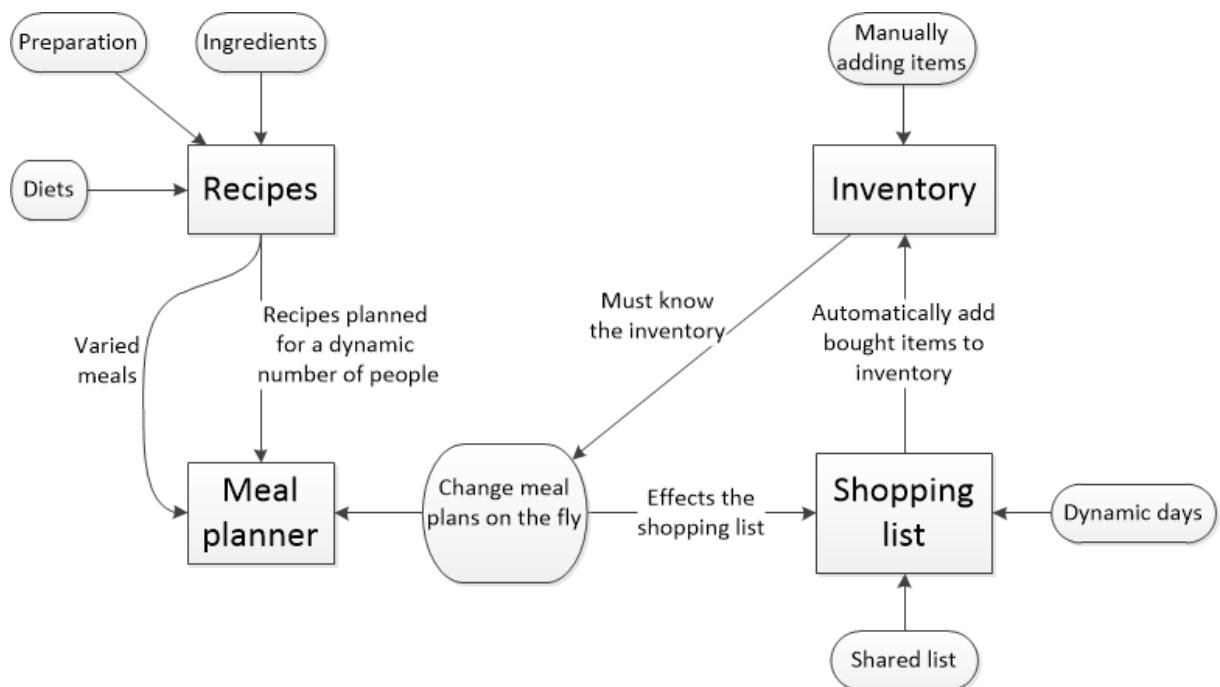


Figure 6.2: Conceptual Model

6.2 Prototype Development

In this section, the conceptual scenarios are made more concrete in the first part of this section, from these scenarios paper prototypes are made through a process where prototypes have been sketched and evaluated, until there was found a final result, which is then shown in the report, in section 6.2.1.

6.2.1 Concrete Scenarios

This section will use the process of specifying design constraints on the conceptual scenarios in section 6.1.2 and produce a number of concrete scenarios. Each concrete scenario has a PACT overview followed by a detailed scenario description, where notes are marked by a set of parentheses, with roman numerals, e.g. (I), these notes are then listed below the scenario, and explain when in the scenario, different design choices can be made.

Concrete Scenario 1

Title: Shopping Without a written list.

PACT overview

- People: Peter. Works at a bank office.
- Activities: Selecting a meal, creating a shopping list and updating inventory.
- Context: in a car and in the local supermarket.

Introduction: Peter is on his way to the local grocery store. On his way he thinks of what he wants for dinner and from this he assembles a list of ingredients that he needs to buy. When he walks around the store picking each item he imagines crossing it off a list. On his walk around the store he picks up a few extra items that he feels is needed at home.

Scenario

1. Peter has just gotten of work and is on his way home in his car. While he drives through traffic he thinks of what meal he wants to eat. He is visited by his sister Laura and her boyfriend John so he wants to make a good meal that requires good and fresh ingredients. He knows that Laura likes chicken(I). Peter therefore decides to buy a large chicken and ingredients for a salad.
2. In the store peter starts by walking over to the vegetable area and begins to check off what he needs to make a good salad(II). He is insecure of what he has at home so he buys all of the ingredients just to be sure(III). He now glances over the vegetables on the shelves and feels like taking a few apples(IV). He Looks down into his basket and checks what he has gotten so far. Enough for a salad and a few apples for the next days.
3. Peter now goes directly over to the frozen products and picks out a chicken. Standing and looking over the products he decides to also buy food for tomorrow(V). He picks up a bag of frozen potatoes and some steaks. He decides not to buy anymore because he thinks there will be some leftovers from to nights meal.

Design Notes

- (I) How does the program enable Peter to specify and find the right recipe that he wants?
- (II) Should the user cross off items on the shopping list or how is it indicated that an ingredient is bought?
- (III) How does the program make is possible for users to see what items that they have at home so they avoid being in doubt?

- (IV) Can the user add items via the shopping list or should they be added directly to the inventory?
- (V) How does the program make it easy for the user to plan a meal at any time e.g. by minimizing the number of steps to plan a meal?

Concrete Scenario 2

Title: Forgetting to buy ingredients.

PACT overview

- People: Jenny, Working as a teacher.
- Activities: Shopping, Checking shopping list and Updating shopping list.
- Context: In a grocery store and in the home.

Introduction: Jenny is out shopping for the next two days with her kids. She wants to shop for lasagna and pizza. When she gets home after shopping she discovers that some ingredients are still missing. She drives back to the grocery store and buys the missing ingredients.

Scenario

1. Jenny is going to the grocery store with her two kids James and Emily. She has not written a list but she does know what meals to **buy for**. She wants to make Lasagna this evening and tomorrow she and her kids are going to make pizza together.
2. In the store she walks around picking up items and checking them off of her list(I). She starts with buying the ingredients she needs for lasagna and then goes on to buy the ingredients to make pizza. Her kids chooses some ingredients they want to put on the pizzas(II) and they are put in the basket with everything else. After buying all the items(III) she can think of for her planned meals she and the kids walk to the checkout.
3. In the car on the way home Jenny talks with her kids about their day. The kids have been in the local sports center so they are very hungry tonight. Jenny quickly unloads the car as they arrive at home. In the kitchen she places the bought ingredients(IV). As she opens the refrigerator and puts in the new ingredients she sees that there are no cheese for the pizzas. She did not buy any when she was in the grocery store because she thought there were plenty at home(V). She tells her kids to stay home and drives back to the grocery store.
4. In the store she goes directly to the dairy area and picks up the cheese she needs. Before she leaves for the check out she thinks of anything that could still be missing. She is not sure there are enough tomato sauce for the pizzas so she picks some extra cans. Now she is sure there is no more ingredients missing and walks up to the checkout. She pays for the ingredients and drives home to cook dinner with her kids.

Design Notes

- (I) Should the user cross off items on the shopping list or how is it indicated that an ingredient is bought?
- (II) Can the user add/ edit recipes in the program with their one variants of the meal?

- (III) Should the user cross off items on the shopping list or how is it indicated that an ingredient is bought?
- (IV) Can the user add items via the shopping list or should they be added directly to the inventory?
- (V) How the program make it possible for users to see what items they have at home so they avoid being in doubt?

Concrete Scenario 3

Title: Receiving text messages to shop after.

PACT overview

- People: Casper, Taxi driver.
- Activities: Shopping and Texting with his wife.
- Context: In the grocery store.
- Technology: Smartphone.

Introduction: Casper is out shopping for the next couple of days. He does not have a full list of all the ingredients that is needed at home. His wife Sandra is texting him and helping him to shop all ingredients that are needed at home. In the supermarket Casper walks around completing his list and updating with what he is told from his wife.

Scenario

1. It is weekend and Casper is in the local supermarket. He has a list of items on his cellphone that he needs to buy(I). Before he starts to shop he texts his wife and asks her if there is anything missing at home that he has to buy(II). He starts by going after the ingredients he has on his phone.
2. When he is about halfway he gets a text from his wife telling him to buy some milk, fruits, eggs, sausage and bread. He adds these items to his shopping list(III). As he walks around collecting each ingredient he checks off each ingredient(IV). He spots some meat on sale and even though it is actually not on the list he decides to buy it anyway.
3. After collecting each ingredient on the list he walks up to the check out. Just before he gets in line his wife texts him and asks if he can get some rice. Luckily he is still in the supermarket so it is not a problem to get the rice.
4. He gets the last ingredient and he now has everything that is needed at home. If there still is something missing it will be bought next time he gets to the supermarket.

Design notes

- (I) What information is shown to each ingredient on the list and how is it presented?
- (II) How would the user know what is missing at home if the wife could not check up on it?
- (III) How does the user add ingredients to the shopping list (add to shopping list or inventory)?
- (IV) Should the user cross off items on the shopping list or how is it indicated that an ingredient is bought?

Concrete Scenario 4

Title: Sending text messages to shop after.

PACT overview

- People: Olivia, Housewife.
- Activities: Checking inventory and making a shopping list.
- Context: In the home.
- Technology: Smartphone.

Introduction: Olivia is in her home when she gets a text from her husband asking if they are missing some ingredients. She looks at his list and checks if they have any of it at home. She also checks if there is anything not on the list that is needed. She texts her husband back with a few ingredients that he can add to his list.

Scenario

1. Olivia is sitting at home and working in front of her PC when she gets a text from her husband. He asks if she can check if they are missing anything from the shopping list he has created at home. She starts by walking out to the kitchen and checks if the refrigerator, the cabinets and their freezer.
2. By looking at what there are in the refrigerator she sees that they could restock on milk(I). She writes it down in a text. Next she goes through the cabinets looking if they got any ketchup, bread, sugar and cereal. There are no ketchup and only half a pack of cereal. She writes these ingredients in the text as well(II). After looking in the freezer she writes that he can buy some frozen potatoes, vegetables and some ice cream. She now sends the text to her husband.
3. After sitting down in front of her PC she remembers that they are getting guests tomorrow. They need some lunch to serve. She writes that the husband should also get some fish, fresh potatoes and fresh vegetables(III). She puts the phone away and starts working again.

Design notes

- (I) Should the user input anything else than name and quantity of the ingredient when adding it to the shopping list?
- (II) Can multiple ingredients be added to the shopping list at the same time?
- (III) Could the user pick a recipe and then the ingredients will be added to the shopping list?

Concrete Scenario 5

Title: Shopping in multiple stores.

PACT overview

- People: Dennis, train conductor.
- Activities: Shopping and editing Shopping list.

- Context: In a grocery store.
- Technology: Smartphone.

Introduction: Dennis has a list of ingredients he needs to buy. He knows that one store will have a sale but the other ones will not. He decides to go to the store with the sale and buy all the ingredients he can get. All the ingredients that he cannot buy at the first store will be bought at another one.

Scenario

1. Dennis has written a shopping list for the next four days(I). The list is quite long so he does not mind shopping in different stores or on two consecutive days. He can see in the sale magazines that one of the local grocery stores has a sale today. He decides to go there and buy as many of the ingredients from his list as he can(II). The rest of the ingredients will be bought in another store that he usually shops in.
2. Later the same day he drives to the store with sales. He looks at his list and starts going around checking ingredients of the list(III). As he goes around he sees some sale offers that he just cannot resist and puts into his basket(IV). When he comes up to the check out counter he looks at his list and double checks that he has crossed off and added the extra ingredients.
3. Dennis buys the ingredients(V) and heads home. At home he unpacks and makes a new list of the items he is missing. The day after he goes to another local store with the intentions to buy all the ingredients on his shopping list. He buys all the ingredients at the store without adding any impulse buys. He pays for the ingredients and drives home to cook some evening dinner.

Design notes

- (I) Can the user specify how many days to shop for?
- (II) How does the user "buy" some ingredients on the shopping list but not all.
- (III) Should the user cross off items on the shopping list or how is it indicated that an ingredient is bought?
- (IV) Should the user input anything else than name and quantity of the ingredient when adding it to the shopping list?
- (V) How is it indicated that an ingredient is bought?

Concrete Scenario 6

Title: Using a written list to shop.

PACT overview

- People: Jack, works at a saw mill.
- Activities: Planning meals and creating a shopping list.
- Context: at home and in a grocery store.
- Technology: Paper and pen.

Introduction: Jack is sitting at home in his living room and planning three days ahead. He is first planning the meals and then he is assembling all the necessary ingredients on a shopping list.

Scenario

1. Jack is sitting in his sofa with the TV turned on. As usually Jack likes to sit down and plan his meals ahead of time. He wants to plan the three next days. He starts by planning the meal for tomorrow(I).
2. He wants to cook a good meal because his sister is visiting. He knows that she love lasagna(II) so that is of course what he will cook. He writes lasagna on his list and as bullet points he writes all the ingredients he needs(III). He does not write from a recipe but only from what he remembers have to be put in a lasagna.
3. The day after he is not sure what he wants. He wants to make something with fish but he does not know if he wants to have rice or noodles to it(IV). He walks into his kitchen and checks what he has. He cannot find any rice or noodles(V) so he decides to make fish with noodles. He adds fish, noodles, vegetables and some spices to his list.
4. On the last day he wants to make a big meal so he can use the leftovers the following days. He decides to make pork chops with rice. He knows that there are no rice at home so he writes that on the list. He now needs to add the pork chops, cream, milk, paprika and asparagus to the list because he knows that he does not have those at home.
5. The meal plan now consists of lasagna, fish and pork(VI). The shopping list has all the items needed to make these meals. The next couple of hours Jack will add items to the list when he discovers what else he needs.

Design notes

- (I) How is does the user add a meal to a specific day in his meal plan via. button, drag?
- (II) How does the user search for specific recipes?
- (III) Are the ingredients for the meal added to the shopping list automatically?
- (IV) Can the user search for different parameters on one search?
- (V) Can the user see if he is missing ingredients for a recipe?
- (VI) Where can the user see an overview of his plan?

Concrete scenarios evaluation

The following text will present the results found by evaluating the different scenario notes. The intentions with performing an evaluation of the scenario notes is to uncover design considerations that has to be addressed. The section is structured so that the domains identified in section 6.1.4, will be used to categorize the different design considerations. If a domain does not have any associated design considerations it will not be used.

Planning

This domain is associated with design considerations from concrete scenario 1, 2, 4 and 6. The considerations are as follows:

1. Enabling the user to specify and find the right recipe.
2. Indication the number of ingredients the user has for a meal.
3. How is a meal planned e.g. number of steps.
4. Making the addition of planned meal ingredients to the shopping list automated by the program.
5. How does the user add a meal to a day on his meal plan e.g. using a button or dragging.
6. How does the user search for a specific recipe?
7. how many criteria or search options does the user have when searching for recipes?

Shopping

This domain is associated with design considerations from concrete scenario 1, 2, 3, 4 and 5. The considerations are as follows:

1. Can the user how many days to shop for?
2. How does the user "buy" some ingredients on the shopping list but not all?
3. What information is shown for each ingredient on the shopping list and how is it shown?
4. Should the user cross off items on the shopping list or how is it indicated that an ingredient is bought?
5. Can the user add items to the shopping list?
6. What information should the user input when adding an ingredient to the shopping list?
7. Can multiple ingredients be added to the shopping list at once?

Inventory

This domain is associated with design considerations from concrete scenario 1, 2 and 3. The considerations are as follows:

1. What information is shown to each ingredient on the inventory list and how is it presented?
2. How is this list kept up to date? can the user add items to the inventory list?

6.2.2 Paper Prototyping

This section includes the processes of envisionment, evaluation and prototyping. The prototype will be a paper prototype which means it will consist of paper sketches depicting how the visual appearance of the program can be made. These sketches are fundamental ideas for the programs design but the design can and most likely will be changed during the development phase.

The navigation between the different windows of the program is done by using a bar in the bottom of the screen. This bar is always visible making it possible for the user to quickly select any of the main screens to navigate to. The bar is shown fig. 6.3.

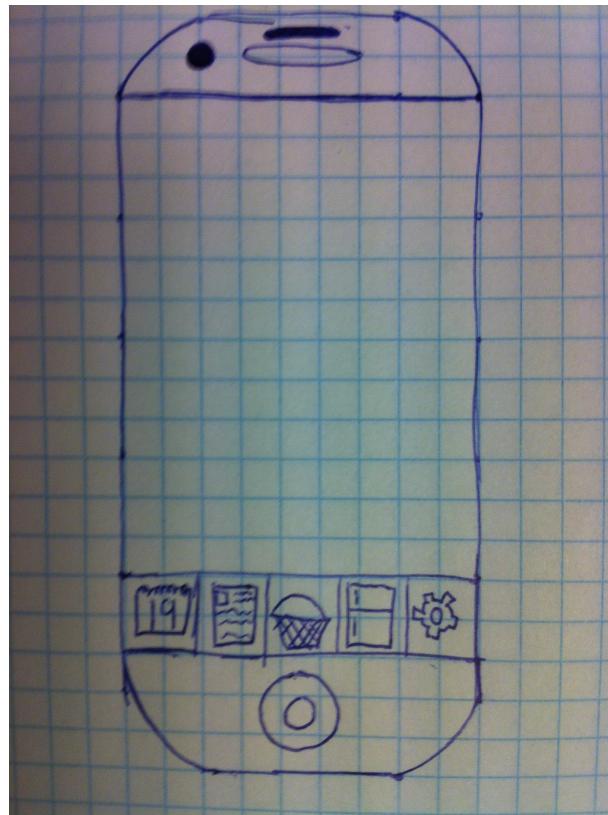
Navigation bar

Figure 6.3: Navigation bar of the program, with the rest of the screen left blank

The navigation bar is placed in the bottom of the screen. The bar has to be readable and its size therefore is quite dynamic as long as the readable requirement is satisfied. Depending on the screen size the navigation bar can fill up a large area of the screen. It has therefore been chosen that the bar will collapse downwards. When the bar is collapsed a line will be displayed at the bottom which the user can press on and the navigation bar will expand upwards.

The navigation bar itself, consist of 5 buttons:

- Meal plan
- Recipe
- Shopping list
- Inventory
- Settings

Each of the buttons take the user to the screen which it represent, and all of these screens will be described later in this section. The icons which are used in the representation are just for the design progress, and not necessarily the final ones.

Meal schedule

The two figures in this section, fig. 6.4 and fig. 6.5, displays two design ideas for the meal schedule. The two ideas both have a weekly and a daily overview. The weekly overview gives the user a quick view of planned meals with little but relevant information. Whereas the daily overview gives the user a specific and detailed overview of specific recipes on a day.

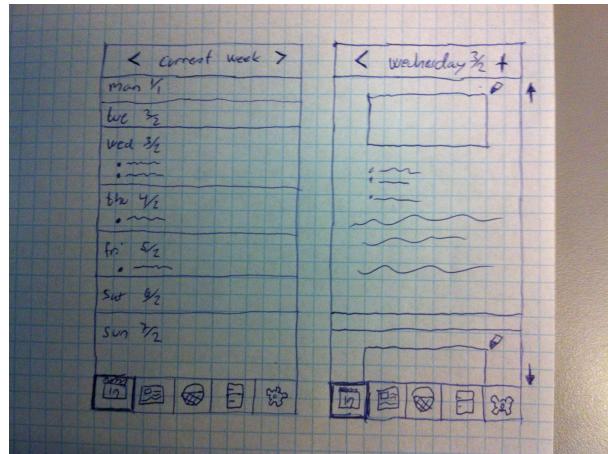


Figure 6.4: The left screen shows a specific week and the right screen shows scheduled meals on a specific day.

Two screen idea On fig. 6.4 two sketches are shown. They indicate the first idea that were thought of for the meal schedule design. The left sketch shows the meal schedule for a week, and the right shows the meals scheduled on a specific day.

Looking at the week schedule (the left sketch on fig. 6.4) there are two elements to look at:

- Top navigation bar
- Schedule

The top navigation bar is used to browse between different weeks. The bar has two navigation buttons and a text block with the **current selected week**. The bottom and the top of the screen in mobile applications are often used to navigate. It is because of this that the top navigation bar is placed where it is. The position also has good symmetry with the bottom bar.

The schedule element of the sketch, shows seven days of the schedule. Seven is chosen because it is a week, and the availability to plan a week ahead is useful if the user plans the same day of the week. Each day of the schedule would show the name of the recipe scheduled, the date of the day, and some of the ingredients needed in the recipe. If the user chooses to click a day, the screen would change to the right sketch on fig. 6.4, and more information about the scheduled meals would appear.

Looking at the specific recipe (the right sketch on fig. 6.4) there are two elements to look at:

- Navigation bar
- Scheduled recipes

The placement of the navigation bar has not changed. Its functionality has changed to suit the new screen. The bar has a button on the left side shaped as an arrow allowing the user to navigate

back to the week schedule. Next to the arrow there are a text block displaying the chosen day e.g. Wednesday 3/2. The last element is a plus icon which allows the user to add recipes to the specific day.

The **scheduled recipes** is a list of all the scheduled recipes for that day. The list is ordered in a downwards fashion, so the user can go through all the scheduled recipes by scrolling through. If the user either want to change the recipe, or delete it, the user can click on the pencil icon, which means edit. A recipe consists of all the information about a recipe, this means picture, ingredients list and the list of instructions.

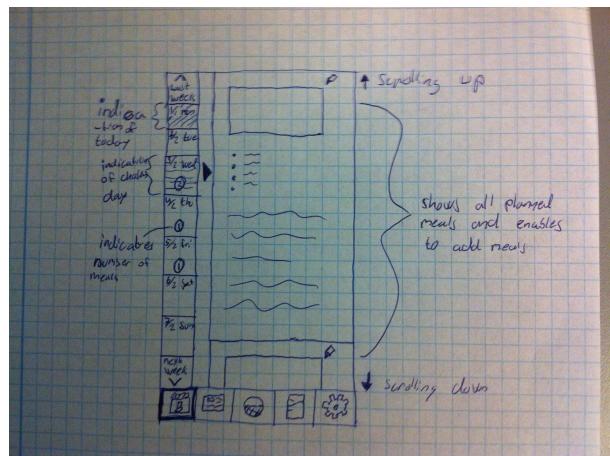


Figure 6.5: This sketch merges both of the screens above into one screen with the weeks on the left bar and the scheduled meals as a list on the right side of the screen.

Combined screen idea In fig. 6.5 another design idea is envisioned. This idea has the weekly overview and the meals scheduled for a specific day combined in one window.

This design idea is used as a concept for the programs meal schedule design. The screen consists of 2 elements:

- Week navigation
- Scheduled recipes.

The week navigation is placed as a bar, in the left edge of the screen. This bar shows 7 days at a time, and have arrows going up and down to go browse between weeks. The reason for arrows and not the scrolling, is that by clicking a day, the scheduled recipes shown will change, so it is easy to misclick, if scrolling by swiping up and down is done, therefore buttons are used. The icons of the week navigation show the date, and a number. The number indicates the amount of recipes scheduled on the day, if no meals are scheduled, no number will be shown.

The scheduled recipes element, works as described as described in the Week Schedule section.

Recipe

In this section the recipe screen is going to be described. Both the functionality of the screen as well as the design principles used in the design process is going to be described. The sketch used to illustrate the use consist of 2 different screens, one screen only showing a full screen of

the recipe, and one screen showing how to browse through the recipe. The browsing screen is divided into 2 sketches, showing the functionality of this screen.

The images sketches referred to can be seen in figure 6.7 and in figure 6.6. The sketch in figure 6.7 is the first screen to see, when browsing recipes. The right screen in figure 6.6 show the expanded version of the recipe browsing screen, and the left shows the full screen of the recipe.

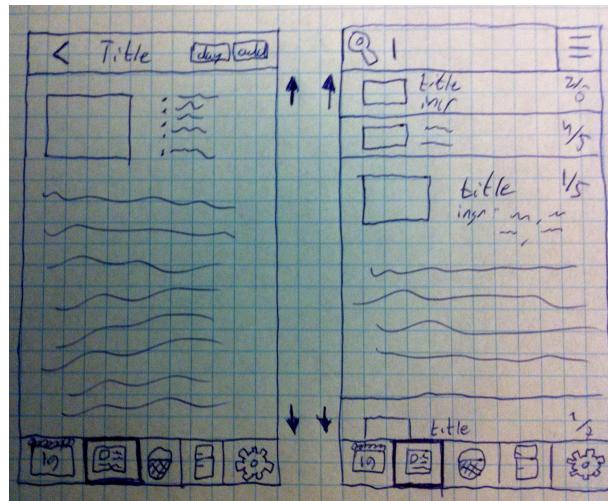


Figure 6.6: One of the final sketch of the recipe browsing screen

Two Screen Idea Looking at the recipe browsing screen (right sketch on fig. 6.6) there are three elements to explain:

- Search bar
- Sort button
- List of recipes

The search bar is located on the top of the screen, as it is where the users will most likely look for it first, as this is where most search bars in mobile applications, but also websites, is placed. When text is entered into the search bar, the recipes containing the text searched for will be shown in the list of recipes.

The Sort button offers the user the functionality of sorting the list of recipes by different requirements. It might be by time to cook, by number of ingredients the user already has in the inventory, or by something else. The sort button is located at the right of the search bar, as this was an easy and convenient place to place it. It only consists of an icon showing that you will be able to sort the recipes by clicking the button.

The list of recipes show the recipes that the user can choose from. These are sorted by the sort button as earlier described, or by the search bar which is also earlier described. If there are more recipes available than the screen can hold, the user will be able to swipe in an upwards motion for more recipes to appear at the bottom and the recipes at the top will disappear. In the right side of each recipe it is shown how many of the ingredients needed to make the recipe the user already have; an example is if the user has 4 out of the 10 ingredients needed the right side will show "4/10".

Looking at the specific recipe screen (left sketch on fig. 6.6) there are two elements to explain:

- Top navigation bar
- Recipe viewing area

The expanded recipe screen can be seen in figure 6.6 on the right. As seen it looks like the recipe browsing screen, but the difference is that this screen show more information about a specific recipe.

When the user has chosen a recipe, the user will be able to click it. By doing so, the recipe will expand, and more information about the recipe will be shown. This will only be the most relevant information, such as ingredients, cooking time, and a picture.

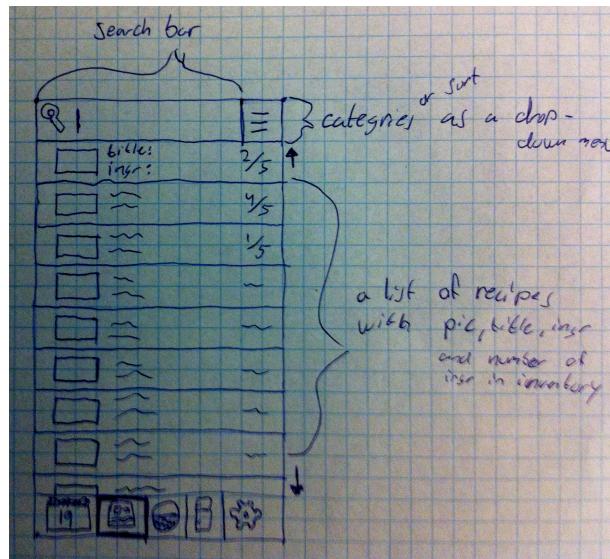


Figure 6.7: One of the final sketch of the recipe browsing screen

Full screen recipe screen The full screen recipe screen is the left sketch in figure 6.7. This screen gives the full overview of the recipe

In the left side, a picture of the recipe will be shown if available, and the ingredients will be on the right side of the picture. If there are needed more ingredients than the picture is long, the ingredients will also appear underneath. After the ingredients list is the instructions to make the recipe. If the recipe is longer than the amount of text the screen can hold, the user will be able to scroll through the rest of the recipe using the fingers.

In the top of the screen is a navigation bar. This navigation bar gives the user the ability to do three things:

- Go back
- Change the amount of people attending the meal
- Add the recipe to the meal plan

The arrow in the left of the navigation screen indicates a go back function, giving the user the ability to go back to the recipe browsing screen if the user did not want to do anything further with the recipe which is looked upon.

The second element enables the user to change the amount of people attending the meal. This would change the amount of ingredients needed in the recipe, and would also update the shopping list, so the user would buy the right amount of the ingredients needed.

The last button give the user the functionality to add the recipe being viewed to the meal plan. This would update the shopping list, and add the meal to a specific day chosen by the user.

Inventory

In this section the sketch of the inventory screen is going to be described. The functionality is going to be described as well as the design principles used in the design process. The sketch consists of 3 sketches of the screen, describing 3 different states of the inventory screen as can be seen in figure 6.8. The first screen from the left, shows the screen when no action has been taken. The second screen from the left shows an expanded ingredient. The third screen from the left shows the screen when the user is searching for an ingredient.

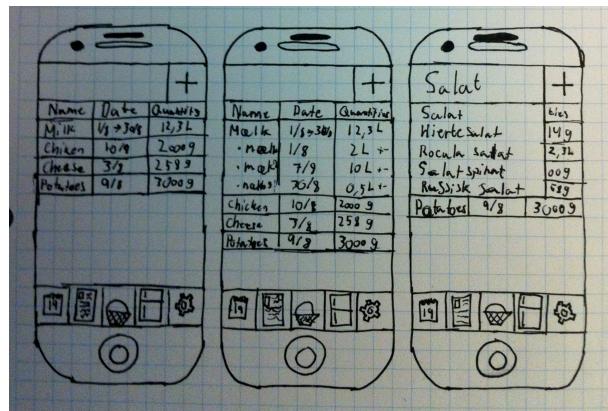


Figure 6.8: The final sketch of the inventory screens

Inventory overview screen The inventory overview screen is divided into two different elements, not taking the general design elements into consideration. These elements are:

- Search bar
- Table

The search bar is placed in the top of the screen, as this is where a user most likely will look for it first, because on most mobile applications and also on websites, the search bar is located at the top of the screen. The add icon is used to make the user able to add items to the inventory, that are not included in the ingredients of the meal plan.

The second element is the table, which is located under the search bar. The table is divided into three columns; name of the item, expiration date of the item, and the quantity of the item.

The first column with the name of the item only holds the information about the items name. The second column shows the expiration date of the item, if the user only have one instance of the item, meaning that there will only be one expiration date, only one expiration date is shown in the expiration date column, but if the user has more instances of an item and therefore different expirations dates, the first and the last expiration dates of the item will be shown, and an arrow in

between the dates will indicate that the expiration dates go from the first to the last. The quantity column holds the information about the quantity of the item.

Expanded ingredient list When the user click on an ingredient, the ingredient will expand and show all the instances of the information as shown on the second screen from the left in figure 6.8. The information is still stored in the three columns described in the inventory overview screen. An instance of the ingredient will therefore show the name of the instance, the expiration date and the quantity of the instance.

Searching for an ingredient The third screen from the left in figure 6.8 shows the search function of the screen. When text is entered into the search box, the items in the search function will expand and lay over the ingredients in the inventory.

The search new overlay will show a fixed number of ingredients which has the searched text in their names, and the user can by clicking one of the ingredients, the expanded search bar will contract, and the item chosen will be shown in the search bar. By clicking the add icon the user will now be able to add the item to their inventory, and if the item already is in the inventory, the item will be an instance that the user can expand to see. The expiration date will then be changed in the table, as well with the quantity.

Shopping list

In this section the sketch of the shopping list is going to be described. The sketch is made up of two different screens, as seen in figure 6.9. One of the screens indicate the overview of the items on the shopping list, and the other screen shows the screen when something has been typed into the search bar.

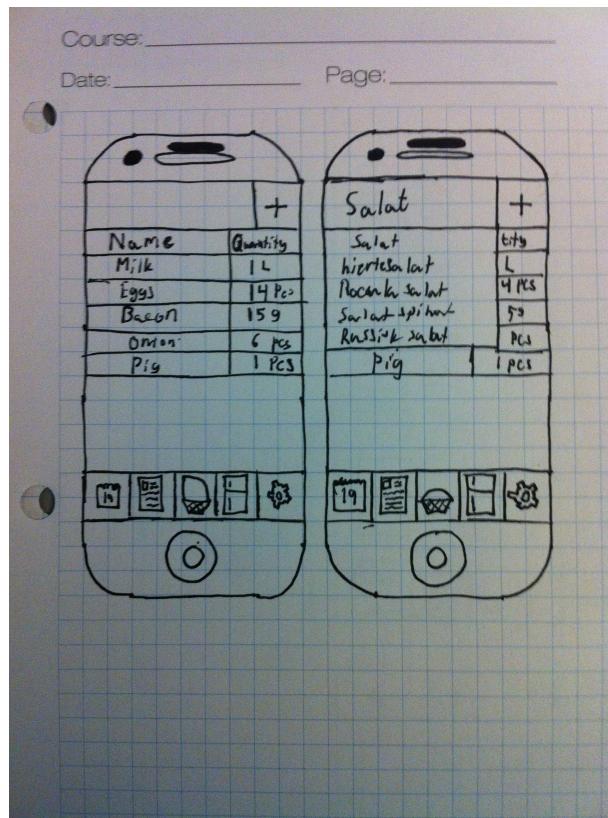


Figure 6.9: The final sketch of the shopping list screens

Shopping List Overview screen The left screen in figure 6.9 is the overview of the shopping list. This screen consists of two elements, not counting the general program elements, the elements is:

- Search bar
- Table

The search bar is placed in the top of the screen, as this is where a user most likely will look for it first, because on most mobile applications and also on websites, the search bar is located at the top of the screen. The add icon is used to make the user able to add items to the inventory, that are not included in the ingredients of the meal plan.

The second element of this screen is the table, which is divided into two columns, one showing the item name, and one showing the quantity of the item. The column showing the name is broader, as the name of some items will be larger than the space needed to write the quantity.

The rows of this screens shows the ingredients on the shopping list, in the sketch, five ingredients is on the shopping list, though there could be as many as needed. If the number of different ingredients fills more than the screen can hold, the user would be able to scroll through the list by swiping upwards. Even though the user would wipe through the ingredients, the first row containing the text "Name" and "Quantity", would still be the first row.

Shopping List Searching Screen The second sketch (right sketch on 6.9) is showing the screen, when a search is performed. The screen will not change, but the search bar will expand and lay over the rest of the screen.

When a search is performed it will show a number of ingredients, in the sketch it is five, and all ingredients containing the word that have been searched for will be shown. Then the user will need to choose the ingredient they want to add to the shopping list, and click on the plus icon to add it to the list.

When a new item is added, **it the user will** need to go down to the quantity field, and put in the quantity of the item that will be needed.

Settings

The settings sketch can be seen in fig. 6.10 and are divided into 2 sketches. The left sketch show the list of settings, and the right sketch show the list of settings, with a specific sketch expanded.



Figure 6.10: This screen displays all of the settings in the program.

Settings List The **setting list** is the left sketch in fig. 6.10 and show a list of settings. The settings can be scrolled through, by swiping up and down, if more settings are needed than the screen can hold. The items shown in the sketch are just ideas for settings, and are just used to visualize the design idea. These are not all settings that will be incorporated in the program.

Expanded Settings List The right sketch in fig. 6.10 show the list of settings, as described in section 6.2.2. Furthermore, this sketch has an expanded setting. This is because if a user click a setting, it will expand, and show more information about this setting. In the sketch, allergies is expanded.

Expanding was chosen, because it would be consistent with the rest of the program, instead of other ideas that where discussed, for example a pop up.

Navigation diagram

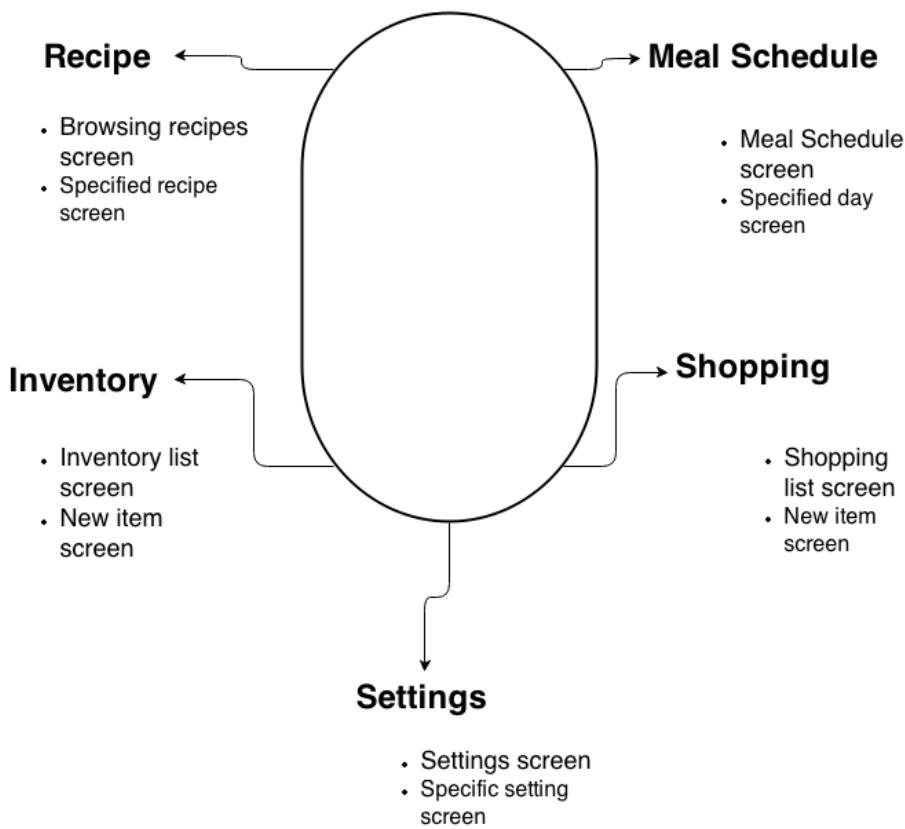


Figure 6.11: This figure shows a navigation diagram for the different screens that have been uncovered in the evaluation process.

Figure fig. 6.11 shows all of the screens that we have found so far through the evaluation of the concrete scenarios. These screens will be sketched in the next section of the report and the content and design considerations of each screen will be specified.



6.3 Design Development



6.3.1 Design Development Introduction

1

6.3.2 Design Language

This section will use the physical design process, in which the Paper Prototyping section 6.2.2 and the Conceptual Model section 6.1.5 are combined, to produce the design language for the program. The physical design process is used to define the look and feel of the program. The process also defines the allocation of functions and the knowledge between a user and a device. The product, which is the design language, is a standardization of the interaction with the program and the physical style of the program e.g. buttons, shapes, colours. This means that the design language is a guideline during the development phase and therefore, that the actual program can differentiate itself from the design language.

Program Screens and Functionality

The domains were identified in section 6.1.4 and they are as follows: planning, shopping, recipes, inventory and general. In Paper Prototyping the domains were used to structure the screens in the program. In the Conceptual Model some of the domains were used as main elements, because of the structural contribution that the domains give, they will be used as a basis for the actual screens of the program. The following text will describe each screen of the program by the functionalities of the particular screen.

Planning By first looking at the Conceptual Model we see that there are some functional requirements stated for the Meal Planner:

- Recipes planned for a dynamic number of people.
- Varied meals.
- Change meal **Plans** on the fly.

By also looking at the Paper Prototyping section 6.2.2, we can add these requirements to the list:

- A weekly overview of planned meals.
 - With information for each meal e.g. name of the recipe, the date for the cooking, and the number of ingredients the user has.
- A view of a specific day and the planned recipes.
 - Display the current selected day the user is viewing.
 - A way to add a meal to the day.

¹ FIXme Note: Skriv intro

- A way to edit already scheduled meals.

Shopping By also looking at the Conceptual Model for this screen we can identify different functional requirements:

- Dynamic number of days to shop for.
- Shared list.
- Automatically add bought items to inventory.
- Meal plan changes effects the shopping list.

By also looking at the Paper Prototyping section 6.2.2, we can add these requirements to the list:

- An overview of all ingredients on the shopping list.
 - Displaying each ingredients name and quantity.
- A search function to find and add specific ingredients.
- A function to buy or add ingredients to the inventory.

Recipes The Conceptual Model also states different functional requirements for this screen:

- Ingredients
 - Displaying name and quantity of the ingredients.
- Preparation
 - An explanation of cooking steps for the specific recipe.
- Diets

The list can be expanded with the functional requirement from Paper Prototyping section 6.2.2:

- A search function.
- A list of recipes with little but relevant information.
 - Information could be a picture of the recipe, recipe name, ingredients of the recipe and the number of ingredients the user has.
- A categorization function
- A view of a specific recipe.
 - Information such as preparation guide, ingredients and a picture.
 - A function to add the recipe to the meal schedule.

Inventory The Conceptual Model can also be used to state different functionality requirements for the inventory screen:

- A function to manually add ingredients.
- Automatic adding of bought shopping list ingredients.

The additions to the functional requirements from Paper Prototyping section 6.2.2 are listed below:

- A search function to find and add specific ingredients.
- A function to remove ingredients.
- A list of ingredients with little but relevant information.
 - Information such as Ingredient name, purchase date and quantity.
- A function or view that easily groups ingredients of the same name but with different quantities and/ or purchase dates.

General The Conceptual Model does not include this domain and the design requirements will therefore solely be identified from the Paper Prototyping:

- An overview of all the different setting categories.
 - Categories such as Stock, allergies, preferences and more.
- A function to expand and display specific categories and the information for that category.

Program Navigation

The main method of the navigation in the program is done via the bottom navigation bar shown on fig. 6.3. This bar can be used from all the screens of the program and directs the user to one of the five main screens written prior in this section, in *Program Screens and Functionality*. If the user navigates away from one of the main screens, e.g. when viewing a list of recipe, the user can press a recipe and view that specific recipe, the top of the screen will then include a back functionality.

Design Principles

This section will discuss different design subjects that are used as general principles in the design language for the program. These general principles are created, to give the program's design coherence and thereby increase the usability.

Navigation The navigation in the program is mainly done via the bottom navigation bar. Consistently displaying and using this as the main way of navigation gives the program better usability. If the user is confused as to where he is in the program he can always use the bottom navigation bar to find his way to a screen that is familiar. Another consistency that the programs navigation has is the use of the top navigation bar. When the user is on any screen that is not one of the main screens the top bar will be used as a backwards navigation. The consistency of using both the bottom and top for navigation functions gives an easy of use and better usability.

Mobile application The program will be designed as a mobile application. It is therefore important that this is considered when defining more general principles for the program. These additional principles are listed and briefly described below.

- General
 - Try to avoid too much redundant information on the screen
 - Avoid too many click function and mouseover functions.
- Text
 - The text has to be readable. That means using logical typography e.g. giving it the right size relative to the screen and an easy to read font.
 - the use of font and text size has to be consistent.
- Buttons
 - The shape and color theme has to be consistent on buttons.
 - The size of the buttons has to be larger enough for a person to click with a finger.
 - Avoid too much text on the buttons. A good idea is to use icons.
 - Avoid using buttons when it is not necessary e.g. When editing an ingredient the program should update automatically.

Colour Choice In order to make the program more appealing, colours are used in the design. It is important that the colours give the right statement, and that they seem attractive to the user.

The colour that was discussed first, using Paletton, was a red colour. This was supposed to be the colour of headlines and such in the program. The reason for discussing red is that red indicates appetite and food [color_psychology]. Red was not chosen though, as it is too strong of a colour, and this might put the user off as it can also indicate danger or alerts.

Since red was not the optimal choice of colour, orange was considered for its similarity to red. Orange has the same quality of red to encourage appetite for the user. Furthermore, orange is a cheerful colour, and it was found to be very appealing, lastly it may be used in the program without standing out too much, or give the user wrong impressions.

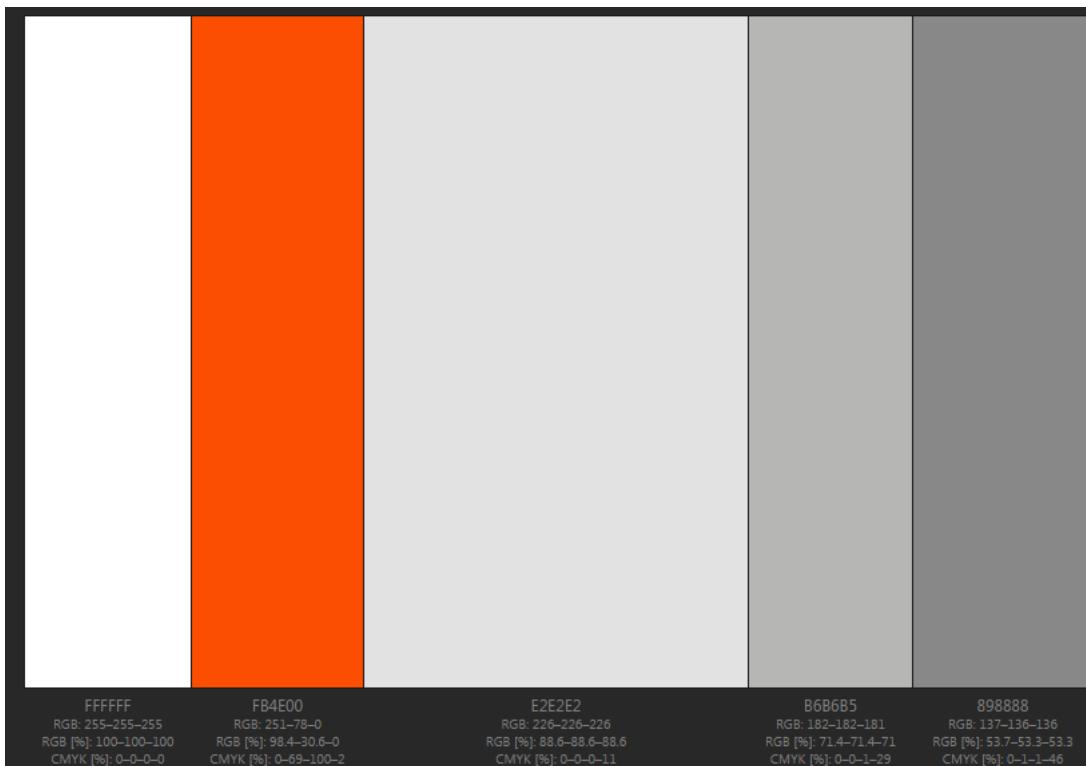


Figure 6.12: The chosen colours for the software

² The shade of orange and a range of gray colours were chosen, and can be seen in fig. 6.12.

The colour #898888 is used for the borders of the program. If something is parted, or a button needs a border, this colour is used. The reason for choosing this colour is that it is much darker than the other colours and gives a clear indication of an ending item.

The colour #FB4E00 is the orange colour as can be seen in fig. 6.12. This colour is the one used for the headline text, as is the main colour of the program, since it is the only colour that can't be characterized as a neutral colour. It is only used for the headline text, as it would be difficult for the user to read the text if it was all orange.

The colour #E2E2E2 is used for the background of the software. This is a very light colour, and it is easy to read black text on it, furthermore does the orange headlines stand more sharp, which makes them easier to see. Therefore this colour was chosen for the background. The white colour #FFFFFF was also considered, but it was too light, and might irritate the user, lastly it did not go as well with the borders as #E2E2E2.

²Fixme Note: cut some of the top of this fig. + maybe put numbers 1 to 5 on the colours, and refer to these, and not only the hashcode

7. Architecture

This chapter describes the system structure in components and processors will be conducted, together with descriptions of the models that has been used.

7.1 Component Architecture

In this section a class diagram will be designed to show the specifications of all complex components. This is done in order to create a flexible and understandable structure of the system.

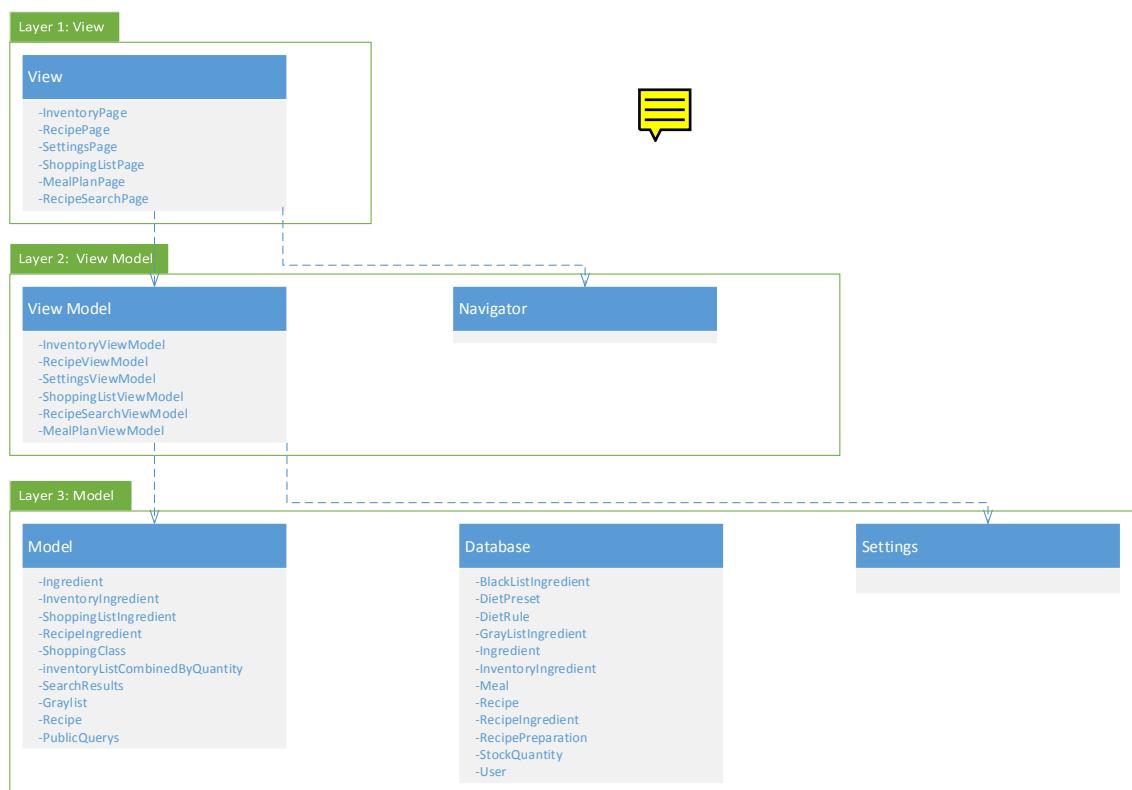


Figure 7.1: Class layer diagram

Figure 7.1 shows the class diagram for the components of the system. The pattern that have been used is the MVVM pattern which is described in Section 7.4.

View The view component spans over the pages(windows) that are used to interact with the user. The responsibility of the View lies in that it should send information to the View Model such that the user can interact with the system.

ViewModel The viewModel component spans over the classes that are associated with the pages in the view. The responsibility of the View Model is to give functionality to the model.

The component uses the classes found in the Model. The **viewModel** also contains a Navigator which is used by the View and ViewModel itself and is responsible for letting the user navigate through the different pages.

Model All the classes that are used in the system is located in the Model component and is used directly by the ViewModel. The responsibility of the Model is to provide classes that can be used to manipulate data and synchronize such data with a database. The objects that can be created through the Model is used in the ViewModel. Another part of the Model is the settings. The settings are saved locally on the machine and is responsible for allowing some personal customization of the local processor.

There are an external standard component which is not shown on the diagram, the component is the Entity Framework which is used to simplify the data modification when it is saved or retrieved from/to the database, Section 7.5 describes the framework in more detail.

7.2 Process Architecture

In this section two distribution diagrams will be shown which explains the processor, the active objects and their connections. All the systems components is on a single processor, but there is going to be multiple processors which is communication with the database. This is due to the system being designed to run on a single device that is not dependent on other devices.

7.2.1 Distribution diagrams

The first distribution diagram will be for the Shopping List.

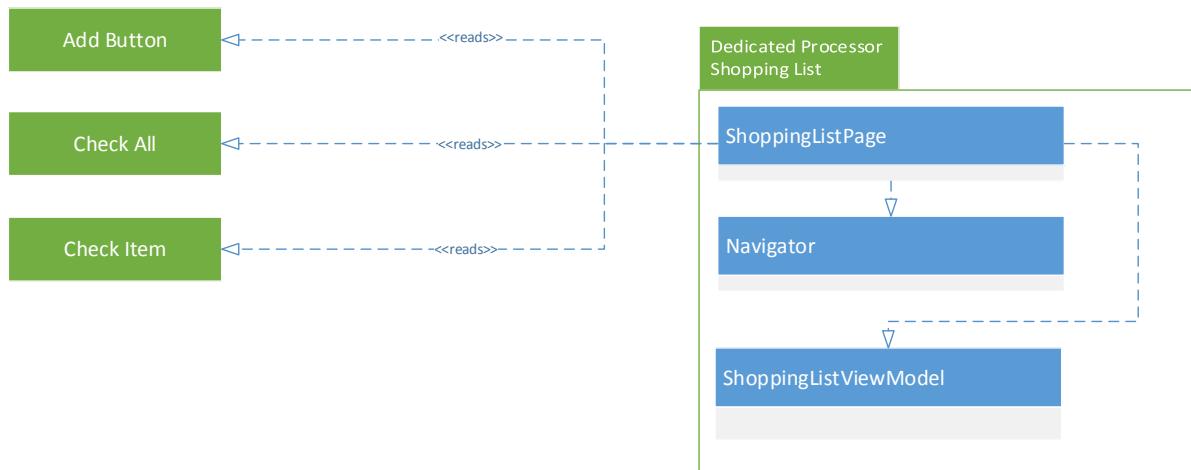


Figure 7.2: The distribution diagram for the Shopping List

Figure 7.2 shows three active objects:

- Add Button - This object adds the checked items on the shopping list to the inventory

- Check All - Checks all items on the shopping list
- Check Item - Checks an single item on the shopping list

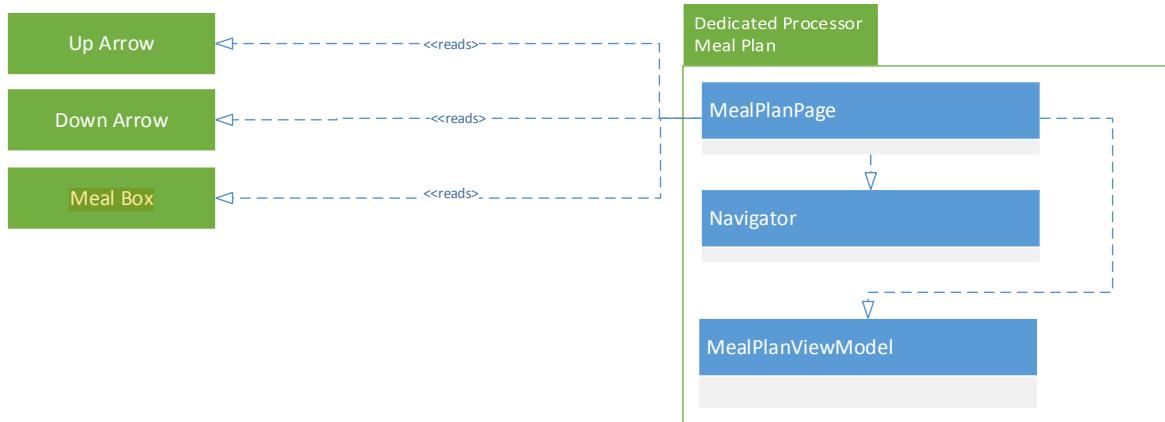


Figure 7.3: The distribution diagram for the Meal Plan

Figure 7.3 shows three active objects:

- Up Arrow - Displays the previous week
- Down Arrow - Shows the next week
- Meal Box - Sends the user to the Recipe Page

The next diagram shows the distribution pattern that have been used.

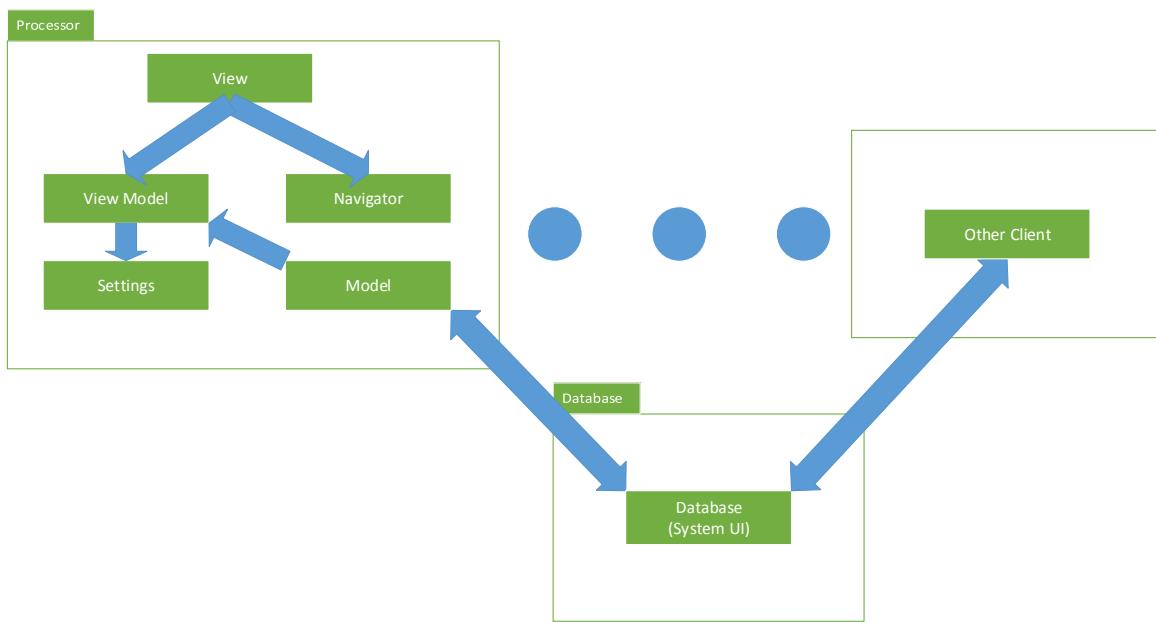


Figure 7.4: The used distribution pattern

Figure 7.4 shows the distribution pattern that have been used, which is a similar pattern to the centralized pattern. Normally the server will perform the functions and keep the model on

itself whereas in this system, the client will perform all the functions and still have some local data which is the settings. A downside to this model lies in the stability, as the system is only functional when a clear connection to the server is established. Another downside to the model is the connection time as almost every interaction with the system will require it to communicate with the server. The server communication creates a natural bottleneck in the system as a server can only handle so much data at any time. When the user searches for a recipe, the entire list of recipes will be looked through in order to acquire the matching results. If it were possible to cash data such as recipes or ingredients, data that does not change often, it would loosen up the bottleneck.

7.3 Database Model / Classes

Figure 7.5 shows a diagram of the relationship between classes in the database model.

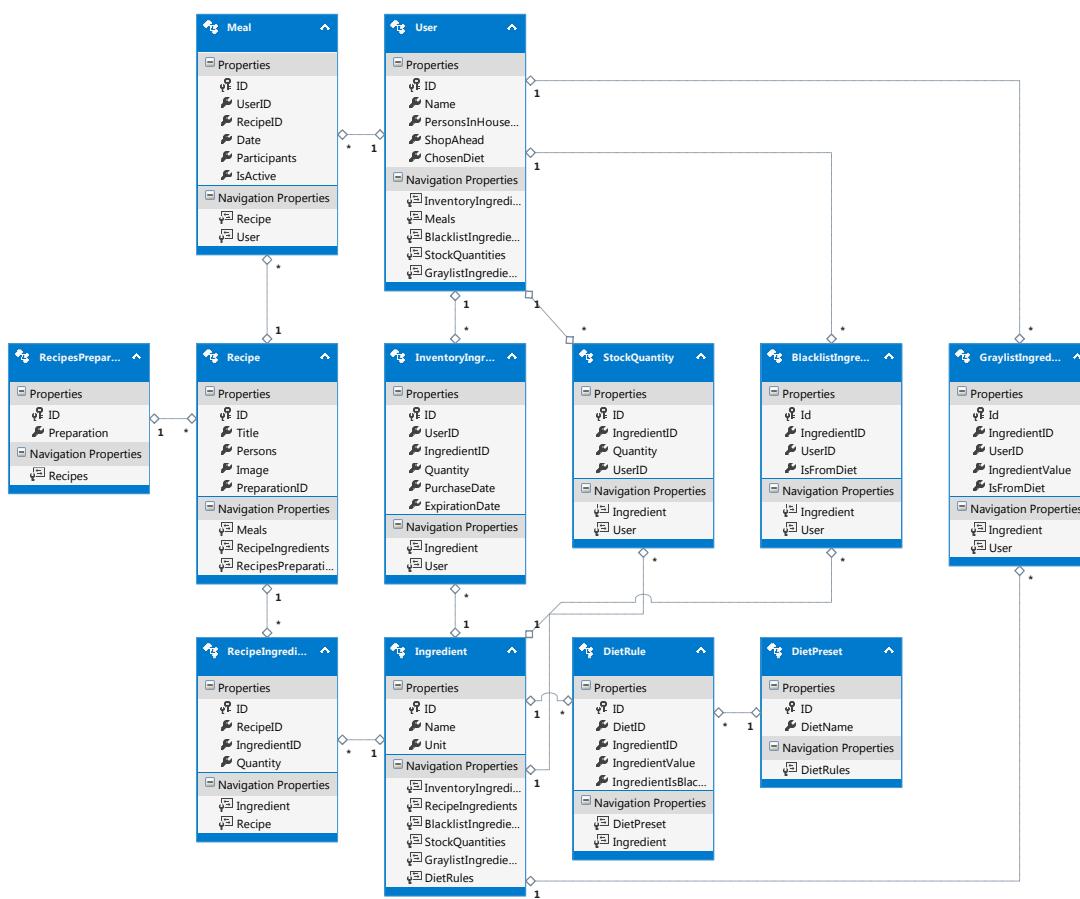


Figure 7.5: Relationship between classes in the database model.

Recipe is a class that describes the individual recipes. The **Persons** property describes how many person this recipe was originally intended for. It has a **onерelates** to the class **RecipePreperation**, by having one property, and to **RecipeIngredient**, by having a list of **RecipeIngredient**.

The **Ingredient** class describes a food and which unit is used to measure it such as kg or ml.

RecipePreparation is a class used exclusively to hold a string property *Preparation* which is a text guide on how to prepare the recipe. The reason this is not just a string property in the Recipe class, is to allow lazy-loading this property from the database when it is needed, thereby allowing the program to list different recipes without having to download a lot of text data that is not displayed.

RecipeIngredient is a class that wraps the Ingredient class to add the quantity property that keep track of how much a recipe needs of the specific ingredient. In the database it also serves a junction table that connects a Recipe with and Ingredient.

The **InventoryIngredient** class describes food that a user can buy or have in the inventory. It is very similar to the **RecipeIngredient** class, but has a *ExpirationDate* property as well, since it is used to describe real existing food, unlike the **RecipeIngredient** class which describes food is needed.

The **Meal** class describes a scheduled meal in a user's mealplan. It contains information about the recipe that defines the meal, how many participants who will be eating, and when it is scheduled for. The *IsActive* property tells if this meal has already been eaten and thereby archived. This usually means that the meal is scheduled for a date within the past.

The **User** class is primarily used to keep preferences about a user, such as a list of **Meals** which is what the mealplan consists of. The preferences include *PersonsInTheHousehold*, *ChosenDiet* and *ShopAhead* which is how many days the shopping list should take into consideration when generating the list. The User is identified with a *Name* as well, but other personal details is not taken into consideration at this time in this project.

StockQuantity is similar to the **InventoryIngredient** class because it contains information about an ingredient and a quantity. It is used as a user preference to keep track of ingredients that a user always want a certain amount of. If the inventory go below this setting the ingredient will automatically be added to the shopping list.

BlacklistIngredient and **GraylistIngredient** is like **StockQuantity** used for user preferences. The **BlacklistIngredient** class contain information about ingredients that the user does not want to appear in a recipe, and **GreylistIngredient** is used to rate and ingredient to adjust it's impact in the search.

1

2

7.4 Model View ViewModel

To separate and organize the application we are using the MVVM-Pattern.

The MVVM pattern is divided into 3 areas:



- The **Model** is the data.

¹Fixme Note: Skriv om DietRule og DietPreset

²Fixme Note: explain why similar classes does not inherit or relate

- The **View** is what you see on the screen.
- The **ViewModel** presents the data in the model to the view.

The pattern has a very loosely coupled design. The View classes does not know that the model exists, while the ViewModel does not know about the view. The Model does not now know the existence and View and ViewModel.

The MVVM is well suited for the WPF platform, since WPF has been designed around the mvvm pattern.

7.4.1 Data binding

Synchronizing the View with the ViewModel happens automatically through data binding. The view binds to properties in the ViewModel, which in turn exposes data from the Model. In fact there is no need to write any code in the ViewModel that updates the view.

The views datacontext is bound to the ViewModel, which makes it easy to bind and display different properties in the.viewmodel on the view.

When a property changes, the new value automatically propagates to the view through databinding. When a user clicks a button on the view, a command in the ViewModel executes the requested action.

The separation between the View and ViewModel makes ViewModel classes easy to unit test. Just like the View, unit tests are just another type of ViewModel consumer??.

7.4.2 Relaying command logic

³ In order to relay commands to private methods...

Updating (INotifyPropertyChanged)

ObservableObject

All the viewmodels inherits from the ObservableObject class, which implements the INotifyPropertyChanged interface, that makes it possible to notify the view about propertychanges

We use a simple library called MVVMFoundation that implements these general purpose ObservableObject / RelayCommand.

7.5 Entity Framework

Entity Framework is a object-relational mapper, part of ADO.NET which are the data-access component of the .NET Framework.

Entity Framework allows developers to access and work with data as objects and properties. A conceptual Entity-relationship model is mapped to a datastore schema model. By using the

³Fixme Note: this is not fully written.

Entity Data Model, the Entity Framework allows data to be treated as entities independently of their underlying datastore representations. This means that instead of manually writing SQL queries, the developer can rely on the object oriented LINQ syntax as well as lambda expressions, directly from any of the .NET programming language??.

8. Design Conclusion



In Scenario-based design chapter 6 the interaction design of the program was defined. This was done by using the Scenario-Based model from the Design and Evaluation of User Interfaces course. This model used real world experiences of users as User Stories section 6.1.1 and from these Conceptual scenarios section 6.1.2 and Concrete Scenarios section 6.2.1 was created. The concrete scenarios was use to develop Paper Prototyping section 6.2.2. The conceptual scenarios and User stories was processed to produce Design Requirements section 6.1.3 and Scenario Corpus section 6.1.4 which was then used to define a Conceptual Model section 6.1.5. The Paper prototypes and the conceptual model were then combined to formulate the Design Language section 6.3.2 for the program. The design language defined a standard for the interaction design of the program.

In Architecture chapter 7 the Component Architecture section 7.1 defined the architecture as a layer pattern (MVVM). The Process Architecture section 7.2 defined the process architecture, by finding the distribution pattern which suited the system best and then adapting that pattern to the actual system. The distribution pattern became a variant of the centralized pattern. Both the component architecture and the process architecture was supported with basic knowledge by section 7.3, section 7.4 and section 7.5.

Part IV

Product Development

9. Pages

In this section the different pages of the program will be described.¹



9.1 Settings

The settings page allows the user to specify different options for the program. The options are categorized as User Specifics, Stock Management, Rated Ingredients, Unwanted Ingredients and Diets.

Under the **User Specifics** category, you can change the how many persons that lives in the household, which is used as the default value for how many persons a meal should be scheduled meals. You can also select how many days you want to shop ahead food for, which means how many days in the future the shopping list takes into account. Finally you can select the default page the program shows when started.

The **Stock Management** category allow the user to add ingredients that they always want to have a specific quantity of. By typing the ingredient into a textbox with auto-completion the user can select the ingredient. In the stock list the added stock ingredients are shown and the quantity can be changed by using numeric up-down control. The ingredient can be removed by selecting them and clicking the remove button symbolized with a minus (-) sign.

Rated Ingredients works the same way as **Stock Management** but instead of setting a quantity, the user can give the ingredient a rating in order to prioritize the search.²

Unwanted Ingredients is similar to **Rated Ingredients** but the ingredients you add will instead be "blacklisted" and thus completely remove recipes with these ingredients from the search.

Diets allows the user to choose a diet, which will make the search prioritize recipes that...³

9.2 Inventory

The inventory page displays a list of the ingredients that the user have in the **household**. Initially the users sees all the ingredients grouped by name, displaying the total quantity of each ingredient and how many individual ingredients there are. By expanding the ingredient with an expander control, the user can see the individual ingredients and change the quantity and expiration date.

In the top the users can add ingredients using the auto-complete control described in section 10.1, which finds available ingredients from the database. The list is initially sorted by name, but can also be sorted by quantity or expiration date.

¹FIXme Note: skriv?

²FIXme Note: explain

³FIXme Note: explain

9.3 Shopping List

The shopping list page shows a list of ingredients that needs to be bought for the recipes in the meal plan for the shop-ahead period mentioned in section 9.1. It displays ingredients and the quantity that is missing from the inventory, according to the mealplan and stock management settings. Each ingredient has a check-box so the user can add the selected ingredients directly to the inventory after shopping.

9.4 Mealplan

The meal-plan shows a week of scheduled meals, and allows the user to navigate to the previous and next week. The weekdays are listed as rows, with seven individual lists of meals belonging to the specific days. By clicking the meals the user can navigate to the Recipe Page to see information about the recipe and/or update meal information.

9.5 Search

The Search page lists recipes sorted by ⁴. In the top it is possible to search for recipes by searching for the recipe name, or comma separated ingredients.

9.6 Recipe Page

The recipe page shows information about a recipe or meal. By clicking a recipe in the search, you navigate to the recipe page which displays the title, image and description of the recipe as well as a list of ingredients. In the top you can add the recipe to the mealplan, and if it is already added you can update the date, or remove it. By clicking a meal on the mealplan you are also taken to this page.

⁴Fixme Note: Describe sorting

10. Custom User Controls



10.1 Ingredient Auto Completion

Since we only allow the use of ingredients which already exists in the database, we need a way to get these ingredients when the user wants to add it in different situations, such as the inventory or the list of unwanted ingredients in settings.

The IngredientAutoComplete is a UserControl that searches through ingredients in the database based on what a user types in to a search field. It will then list the most relevant beneath the search field and allow the user to click one.

It exposes the DependencyProperties; MaximumItems and SelectItemCommand, which can be used to set how many items the should be displayed, and which command should be executed when the user selects an ingredient. This is set on the pages that uses this control.

When SearchText property changes the control will try to repopulate the list of found ingredients. If a previous search with less letters came up with fewer ingredients than the maximum, the control will not attempt to query the database again, since that could not possibly give more results.

11. ViewModels

All pages has a corresponding ViewModel which it is binding its data-context to. So whenever data changes the View will be notified, and the ViewModel will receive input from the view.



11.1 Mealplan

The MealPlanViewModel has a DateTime property called ActiveDate, that is used to reference which week the user wants to look through meals. It has 7 other DateTime properties which references the individual days in the week. When ActiveDate is being set, it updates the MondayDate property, which is used by the remaining properties as reference.

It also has a collection of Meal's for each day in the week. The collections contains the meals that belongs to the different weekdays according to the active date.

These meals are added to the collection when changing the ActiveDate property? ¹

11.1.1 Methods



When clicking the arrow buttons to navigate between weeks, the button invokes a command, either NextWeek or PreviousWeek, which changes the ActiveDate property, and updates the Meal collections.

11.2 Recipe

The RecipeViewModel has a Recipe and a Meal property which are set, on initialization.

`public RecipeViewModel(Recipe recipe) public RecipeViewModel(Meal meal)`



If initialized with a Recipe object, the Meal property remains null, and page only displays information about the Recipe. If the Meal is set either on initialization or when adding a Recipe to the mealplan, the page will also display information about the meal, such as date and participants.

The ViewModel has the commands AddUpdateMealCommand and RemoveMealCommand. The AddUpdateMealCommand will either create a new meal with the recipe and add it to the mealplan, or update the meal if it is already set. The RemoveMealCommand removes the meal from the mealplan, the button that invokes this command is only visible when the Meal property is set.

¹FIXME Note: Describe this

11.3 Inventory

The InventoryViewModel has a property called `InventoryIngredients` which is a collection of all the food that the user has in the inventory.

The CollectionView of `InventoryIngredients` is given a `PropertyGroupDescription` which groups all ingredients with the same name as one item. The name and total quantity can then be displayed on the page, while the individual ingredients can be expanded as well, as described in section 9.2.

The CollectionView also has a `SortDescription` which sorts the list by `Name`, `ExpirationDate` or `Quantity`. The chosen `SortDescription` can be changed by changing a property called `SelectedSortIndex`. On the `InventoryPage` this is currently done with a `ComboBox`.

The ViewModel also contains commands for removing and adding an ingredient.

11.4 Settings

The SettingsViewModel `has` exposes several properties for setting different options for the program.

The integer properties `PersonsInHouseHold` and `ShopAhead` can be adjusted for the current user through databinding.

The property `SelectedStartUpPage` is used for selecting which page the program should display when started. This is the only setting that is not saved in the database, but as a option saved on the device instead, since it is setting for the program and not the user.

The Settings page allows to add ingredients to different lists such as stock and unwanted ingredients, see section 9.1 for more details. The following properties is used to keep track of the selected ingredients in the different lists when adjusting settings:

Listing 11.1: Selected list items

```

1 public BlacklistIngredient SelectedBlackListIngredient { get; set; }
2 public GraylistIngredient SelectedGreyListIngredient { get; set; }
3 public StockQuantity SelectedStockQuantityIngredient { get; set; }
4 public DietPreset SelectedDietPreset { get; set; }

```

Most methods in the SettingsViewModel is used for adding and removing ingredients to list, and incrementing or decrementing values.

12. Navigator

The MainWindow has a Frame control that shows the content of the different pages. In order to navigate between the different pages and still separate the Views from the ViewModels, we created a special class called Navigator. This is a static class following the singleton pattern.

When the application starts the Frame control of the MainWindow is tied to the navigator. The Navigator can not be instantiated, but Views and ViewModels can access its static navigation commands. It has a command for navigation to each page. Listing 12.1 below shows the GoToInventoryCommand.

Listing 12.1: GoToInventoryCommand

```

1 public static ICommand GoToInventoryCommand {
2     get {
3         return new RelayCommand(() => Navigator.Navigate(new InventoryPage()));
4     }
5 }
```

The public ICommand property GoToInventoryCommand, relays its functionality to the Navigate method by passing a new InventoryPage as parameter.

The Navigate method shown in listing 12.2 routes the parameter to the NavigationService of the Frame control on MainWindow.

Listing 12.2: Navigate method

```

1 private static void Navigate(Page page) {
2     NavigationService.Navigate(page);
3 }
```

The above example has been simplified and leaves out null checking, compared to the actual implementation.

Furthermore the Navigator class also contains a list of all pages in the program corresponding with a title, which is used on the settings page, when choosing a start-up page.

13. Testing

When the program had been made, some testing where done to find problems within the program. The test included people outside the group, and where conducted with an objective view. The tests ended up with a list of requirements for the program, that needs to be changed in order for the users to get the best possible program experience.

13.1 Usability Test

To test the usability of the program, a usability test were conducted which included people outside the group, to achieve an objective standpoint of the usability. The usability test was an **IDA** (Instant Data Analysis) test.

13.1.1 Conduction of the test

The setup for the test consisted of a video camera filming the test participant and test monitor, a screen recorder recording what the test participant did within the program. Two people where describing what the test participant did, and a test leader was explaining what the program did, making sure the participant was doing to the task, and helping if the participant got stuck.

When a participant was going to be tested, he was first explained what the program did, and what problems it was supposed to solve. After this the participant was presented with the tasks that he was supposed to do within the program. They were told to read the tasks out loud, so when reviewing the video, it was possible to know which task the test participant where executing. The tasks that the participant did can be found in appendix E.0.3.

When the participant had completed all of the given tasks, some follow-up questions were asked. The questions was asked to see if the user had any comments that was not mentioned throughout the execution of the tasks. These questions can as well as the tasks, be found in appendix E.0.3.

Five test participants did the tasks, and after all of these were through the test, the analysis could begin. The point of the analysis was to find common errors that the participants had within the program, so these could be changed. The errors was then divided into categories of the following:

- Critical problem: Problems that must be changed for the program to work probably.
- Serious problems: Problems that can be changed for a better usability.
- Cosmetic problems: Small problems that does not need to be changed, though changing them will provide a better usability.

13.1.2 Test outcome

When the test had been conducted the problems was identified, the problems was summed up into the following list:

- Test participants clicked the day they wanted to plan a meal in the mealplan page, as they found this to be the most intuitive way to plan a meal, instead of going to the search page.
- Some test participants found it more intuitive to change the "number of days to shop for" within the shopping list page instead of the settings page. Some of the participants found that a link from the shopping list page to the settings page would be sufficient to fix this problem.
- Participants found it hard to know that they needed to click the update button for a planned meal to be updated, when changing the date or the number of meal participants.
- Some participants did not know that a meal was deleted from the meal plan, after clicking the subtraction sign, there was not enough feedback.
- Some participants mentioned that they had a hard time finding consistency in whether the data was updated automatically or they needed to click an update button.
- Some participant mentioned that they had problems seeing how the settings screen were divided. They mentioned that there was too much space between some of the objects within the area of a setting.
- When scheduling a recipe, some of the participants mentioned that the top bar was hard to figure out, and the information in this top bar needed to be communicated clearer.
- Some of the participants mentioned that there was not enough feedback when adding and ingredient to the rated ingredients list.
- Some of the participants thought that having a rating of 100 was too much and a rating of 10 would have been better, where as other thought that 1-100 was a simple and easy scale to use.

The list of problems was then analysed and the problems were divided into the categories of critical, serious, and cosmetic.

Critical Problems

- Not knowing to click the update button when changing information of a scheduled meal.
- Not knowing that a meal was deleted after clicking the delete meal button.
- Consistency of whether the program updates automatically or if the user have to click to update.

The first critical problem can be solved by letting the program update automatically. The program updates automatically all other places in all other screens, so by letting it update a recipe the moment something is changed and not by clicking an update button, would be come natural for the user.

The second critical problem can be solved by exiting the recipe screen and going to the meal plan, when deleting a meal. This way the user could see that the recipe was deleted, because it would not be shown in the screen any more, and it would not be shown in the meal plan either.

The last critical problem can be solved in the same way as the first one. The only place in the program that does not automatically update is the changing of a recipe, so by making this update automatically, consistency throughout the program would be achieved.

Serious Problems

- Trying to plan a meal by clicking the day in the mealplan page, instead of going to the search page.
- Finding it hard to know how the top bar in the recipe screen worked.
- Changing the number of days to shop for within the shopping list page instead of the settings page.

The first serious problem can be solved by implementing a method that lets the user go to the search page when clicking a specific date, and when the user then **clicks finds** the recipe they want scheduled, the date that it is set to by default should then be the date they clicked in the mealplan screen.

The second serious problem can be solved by making clearer icons. The participants of the test had no problems with deleting a recipe as the subtraction sign is a common sign for deleting something, the same goes for adding a recipe as the addition sign is a common sign for adding something. In the critical problems it is described that the program should update automatically in the recipe screen, so the update button could be deleted. If a label with the text "Participants" were placed left for the participants box, then the user would know that these where participants, and in this way the second serious problem could be solved.

The third serious problem is easily solvable, as this can be done by moving the ability to change the number of days to shop for to the shopping list page instead of the settings page.

Cosmetic problems

- Finding it hard to see how the setting screen were divided.
- Not getting enough feedback when adding an ingredient to the rated ingredient list.
- Having a 1-100 rating is too much, 1-10 is enough.

The first cosmetic problem can be solved by deleting some of the blank space in the settings screen. An example could be to move the buttons up by the side of the listview, as the buttons take up all the horizontal space it is in right now.

The second cosmetic problem can be solved by letting a label say when the item has been have been added to the rated ingredients list.

The third cosmetic problem can be solved by replacing the rating system by 10 star rating system.

13.1.3 Conclusion

The program has usability problems that can be solved. These problems are divided into all the three categories, critical, serious, and cosmetic. The problems are equally divided into the three categories, so some of the problems are important to fix if the program was to be released to the public.

Part V

Summary

14. Delimitation

In this section the delimitation of the program is done. When the report was written and the sources were found, a lot of good ideas for the program was discussed. These ideas were then written in the report, so to make the program, it was needed to delimit from some of the ideas, as the program might either become too complex or too time craving to make.

14.1 Not in program

In section 1.3 it is mentioned that incorporating leftovers is a good way to avoid food waste, according to The Danish Ministry of Environment. This is partially done in the program. In the program, when the user prepares a meal, the ingredients used will be taken away from the users inventory. If the user have leftover from the meal, they can add them to the inventory, and if these leftovers are needed to make some of the recipes incorporated in the program, these will then show up higher when meals are being planning.

In section 1.3 there is another initiative to reduce food waste, and that is by making stores available to sell quantity discount, but for the user to pick the items up at different times. This is not incorporated in the program, as this would require the project to have one additional focus point than meal planning.

In section 1.3 another suggestion for reducing food waste is to inform the user of different methods to see if the food has gone bad. This is not done in the program. The program does keep track of the expiration date of the items in the users inventory, but does not inform the user of different ways to see if the food has gone bad, as some of these dates might be guidance and not actual expiration of the food. It was suggested to to colour the food information text in the program when an item expired, this was never implemented in time.

In the section 1.3 it is also mentioned that if it where possible to share quantity discounts with other people, food waste would be reduced. This is not incorporated into the program because the focus of the report is on meal planning, and therefore it would require a complete shift of focus to make a program that was able to do this.

By keeping track of what is being thrown out, it can be seen if the same type of ingredients are being thrown out often, and the user might need to buy less of this ingredient it is suggested in section 1.3. This is not implemented in the program, as the program only tells the user to shop for the items needed for the recipes, and therefore the shopping list will only consist of needed items.

It is mentioned in section 1.3 that by being creative with the leftovers, and making meals, from leftovers of which is already parts of a meal, food waste can be reduced. This is partially incorporated in the program, as the user can put in the leftovers in the inventory. If a recipe uses these leftovers, recipes including these will be listed higher, when the user searches for a recipe to schedule. Though there is not a specific initiative to use leftovers from meals creatively.

In section 2.1 it is mentioned how a conflict can arise when multiple users try to change the same meal plan. The function of multiple users using the same meal plan, has not been incorporated in

the program though. The reason for this is that it would require to expand the program, and that time for this was not available.



Section 2.2.1 has some ideas for the program. Not all of these were used. The section suggest the availability to create custom recipes through the program, or buying recipes for the program as a form of program expansion. This feature has not been created for the food planner program, as the recipes in the program are taken from a website??, and are freely available on for the public to use.

In section 2.2.1 it is also mentioned that a user should be able to register for an account to synchronize between different devices. This is not incorporated, as the time for the creating this feature was not available.



In section 2.2.1 it is mentioned that a "surprise function" is an idea already implemented by other meal planning services. This feature creates a meal plan for the user, without the user needing to plan it themselves. This function is not incorporated in the program. However, there are recommended recipes in the search page based on the users information.

Section 2.2.1 also suggests the user being able to exclude recipes that takes a specific time or more to cook, so the user can choose to cook recipes that require a specific time or less. This is not done in the program, because the website where of the recipes have been taken does not include the time it takes to cook the recipe.

Section 2.2.1 mentions that by providing the user with inspiration for leftovers, food waste can be lowered. This is not incorporated in the program, though the user can add leftovers to the inventory list, if the leftovers ingredient type is present in the database.

To have featured meal plans for the user to gather inspiration or for the user to choose from is mentioned in section 2.2.1. This is not made in the program, as the program only contains one meal plan. The user can get the inspiration from the recipes screen.

Customer plans is an idea mentioned in section 2.2.1, it means that the user should be able to browse through other users food plans for inspiration. This is not incorporated in the program as there is no online community made available in the application.

"My binder" is an idea mentioned in section 2.2.1 which suggests that the user should be able to bind recipes they like, so they can find them later easily. This is not in the program, as the project is used to lower food waste by giving the user the ability to find the recipes with the most ingredients already owned.

14.2 In program

In section 1.3 it is also suggested that by following recipes and using the exact amount of the ingredients will reduce the food waste. This has been incorporated into the program as a user has the ability to change how many people the recipe is for, and then the amount of the ingredients needed, also changes.

In section 1.3 it is also suggested using the items that expires soonest. This is part of the program, as the searching of adding a new recipe to the meal plan takes into consideration, and sorts the recipes so that the ones with ingredients expiring sooner gets higher up.

In section 1.4.1 it is mentioned that people wanted realistic food, not food from cookbooks, because of the time and resources needed for making those. This has been taken into consideration in the program. The recipes from the program are stripped from a website, these recipes vary in time and easiness to make. The searching when planning a meal shows how many of the ingredients you already own, so you do not need to go shopping for a lot of ingredients maybe to make a recipe, as you can see if you already have what is needed to make the meal.

In section 1.4.1, four criterias was found when people were making a meal plan: Food you already have, food you like, price, and stock of stores nearby. This is partially incorporated in the program. As described previously, food the user already own is considered when planning a meal. In the settings page you can rate ingredients, so you can choose ingredients you like more, and they will be prioritized higher, when searching for meals to schedule on your meal plan. The price of the food, could not be obtained, as no available database where found which included prices for ingredients. Stock of stores nearby was not possible either, as supermarkets do not let their stock **be** available to the public.

Section 2.2.1 has a list of ideas for the food planning program, that are inspired by other programs. One of the ideas is the inventory that keeps track of the ingredients a user need. This is included in the program as Stock Ingredients.

In section 2.2.1 it is also mentioned that the availability to create meal plans is an idea for the program. This have been made in the program, as the user are able to create a meal plan, that meals can be scheduled in. This is combined with the "Planned Meals" idea mentioned in section 2.2.1.

Section 2.2.1 also mentions how the user should be able to add groceries to a shopping list. This is a feature in the program, as the items needed for the recipes are added automatically to the grocery list.

Section 2.2.1 presents the idea of the user being able to plan both breakfast, lunch, and dinner on the same day. This is incorporated in the program, as the user is able to plan as many meals as needed.

In section 2.2.1 it is mentioned that the user should be able to choose some dietary preferences. This has been made in the program, as some test diets have been set up, so the user can choose the specific diet he/she is following. The rating system and blacklisting system also allows the user to choose which ingredients he or she really like to use or do not want to incorporate in their meal plan, which allows for the user to make a diet more specific.

15. Conclusion

It is a problem that a consumer wants fresh groceries, and at the same time purchases too many groceries, because they do not want to do much shopping. This problem is solved through the software solution because the user will not buy too much food, because the solution only tells the user to buy as much as is needed for the recipes scheduled.

It can be hard to plan a meal schedule for many people, but through the software solution the meals can be scheduled for more people, and the shopping list will be **update** accordingly. This is also a trait for people living alone, since they on average have a higher food waste level than people living together and therefore the ability to only use a specific amount, will both lower their food waste level as well as the cost of the meals.

Having a diet can be a problem when going shopping, because it can be hard to find the specific ingredients needed for the specific recipes, by having a shopping list from home, which includes the ingredients of the recipes, the user will much faster be able to find the ingredients.

By not knowing what you already have at home, the food waste level is bound to rise to be higher than if you are keeping track. The software solution allows the user to keep track of everything in an easy manner, by adding items either directly to the inventory list, or by adding them from the shopping list after the user have bought the item.

The way that the food waste level have been lowered through the software solution, is by letting the user make a meal schedule, as this have been showed to lower the food waste level. The functionality of having to shop only for the amount of what you need, also lets the user get a lower food waste level, and keeping track of the inventory makes sure that items already owned are not bought.

The program have the ability to let the user choose a specific recipe that they want to follow, these will put items that can not be eaten on the blacklist, making sure that the recipes that the program lets the user schedule follows a specific diet.

16. Future Improvements

In this chapter the improvements which could be made in the future will be listed and described. A reflection on how the program could be used in other contexts will also be made in order to explore new areas.

16.1 Improvements

The improvements mentioned in the Not in program should all be included in a full version of the system. These changes should be accompanied by the changes presented in the Usability Test.

More changes have been thought of while developing the system and writing the documentation. These changes or improvements will be described in this section.

An improvement to the system would be to include a budget page where the user could manage his or her spendings for the month. It should be possible to set a budget for any month, and the program should then generate a meal plan that will fit to the budget. The shopping list should display the total cost of the items present on the list. This change would be beneficial as some users have applications on their smartphones or computers that help them with their budget. By having a part of the user budget in the program, it becomes more of a complete package for the user.

Something else that was thought of would be the possibility to pick or exclude shops to shop from. When the user is browsing their shopping list, they should be able to get recommendation on where to buy the different products. Some of the shops might lie on the users route back from home or their children's school. This feature could be expanded to allow the user to set a maximum distance for how far they would like to travel when shopping. If they set a distance of four kilometres, the program will only look for shops within a four kilometre radius from their home. If the needed ingredients on the shopping list could not be found in the shops, the user should be informed on the problem and recommended other shops that lies outside the radius.

Another considered feature were to include "smart shopping". The system should look for sales and try to incorporate them into the shopping list so the user would only buy the items when they were on sale or if they were needed soon. Normally the shopping list would only show the items that are within the "days to shop for" time limit, but if an item that are needed later comes on sale, it should also be added to the list. This goes together with the two previously improvements as it becomes easier for the system to stay within the budget, and it allows it to only look for sales in certain shops.

16.2 Reflection

The programs focus could change to be on bigger groups of people instead of families. A kindergarten could implement the system in order to track what the children should have for lunch. Some of the children could suffer from milk allergy or diabetes and would therefore need

special diets. The program should allow the **teachers** to make appropriate servings of lunch to the children, while helping them come up with recipes for alternative dishes. This would make the administrative work easier and minimize food waste at a place with many individuals. In order to make such a system new recipes have to be implemented as the current system is focused on dinner recipes and not lunch recipes. The recipes should also be focused on children and should promote healthy dishes.

The program could also be redesigned to **be website** for a catering company. The customer would be people who would order food to their parties. They would do so by going to the website and then choose the **dished** they would like to order together with the expected guest count. The catering company would then get an order with a list of recipes they would have to cook together with an auto generated shopping list and price. The price would be based on the shopping list and other factors such as how time constrained the dishes would be to cook or **alike**. This could double up as being an administrative program that shows the appointments they would have with their customers. This would help the company save time by automatically making shopping lists and scheduling their calender. The food waste aspect would not be as important, as the amount of food made would probably not be changed by using the system. However, it might help the company to not buy unwanted ingredients, as the system would keep track of what there is currently **stocked**. The obstacles that would arise if this system were to be made would be to convert the entire program to be a website instead, even though some of the current features would not be ported. The interface and how to navigate would need to be changed entirely as there would be a separate customer view for the customers, and an administrative view for the staff of the company.

Part VI

Appendix

A. Interview answers

A.0.1 Interview answers

25 year old single male

- Sex: Male
- Age: 25
- Education: Vocational
- Work: Skilled Sailor
- Household: 3 singles

What do you characterize as food waste, and is it a problem in the household?

If any kind of food is thrown out, it is food waste according to the participant, furthermore he thinks food waste is a problem in the household, he elaborates that he thinks it is because some food is sold in bundles, that are too big compared with what is needed, and that they therefore buy too much food.

Is there a difference in the type of food, that is thrown out?

It is mostly vegetables, which is thrown out in the household.

What factor is in focus, when food is thrown out?

The participant only look at the freshness of the food, and does not take best before date in account.

Planning meals and shopping.

The participant does rarely plan more than one day ahead, and does not plan at any specific time during day. Furthermore the participant mostly shop for what is wanted the specific day, and it is rarely considered, if a meal could be made from what is in the home. No one specific in the household do the shopping or the cooking, but they try to schedule the shopping with who has the time, or already has to be out.

What is the most important about your diet?

Quality is the most important, closely followed by organic food, they try to avoid food waste, but it is not in focus.

Do you use leftovers?

The participant always use leftovers, both because he finds it stupid, to throw it out, but also to save some money.

When do you shop, for how long, how much and how many times?

Preferably in the morning, because there are less people. The shopping only takes 5-10 minutes and is done 4 - 5 times a week. Breakfast and lunch is often bought for a few days, where the dinner is mostly bought at the day it is needed.

What affects you in the shopping situation?

The participant is not affected by sales in general, but do impulsive shopping if groceries are on sale, because they are close to the best before date.

53 year old married female

- Sex: Female
- Age: 53
- Education: University
- Work: Pedagogue
- Household: 2, one married couple

What do you characterize as food waste, and is it a problem in the household?

Sales that requires you to buy to large quantities, which makes you buy to much, is seen as food waste, together with bad usage of leftovers. Food waste is not seen as a problem in this household.

Is there a difference in the type of food, that is thrown out?

Even though food waste is not seen as a problem, it rarely happens that bread is thrown out, this is because the packages the food is sold in, are to big.

What factor is in focus, when food is thrown out?

The participant only look at the freshness of the food, and does not take best before date in account.

Planning meals and shopping.

The participant plan in the way that, food is taken out of the freezer in the morning, furthermore are there often made extra food, to have easy and fast prepared food, for the next few days, or to have something for a lunch, to bring to work. What needs to be shopped is mostly planned in the morning, when the food is taken out of the freezer, and the participant therefore knows what needs to be bought, if something unexpected happens, resulting in an unexpected dinner, makes shopping after work, a necessity. It is mostly the participant who does the shopping, but the other person in the household, may be asked to do some shopping.

What is the most important about your diet?

The participant want to avoid food waste as a primary focus, next is the prize and the excitement of the meal, taken into consideration.

Do you use leftovers?

The participant always use leftovers. And plans a dinner, so there will be leftovers, to ease the cooking for days to come.

When do you shop, for how long, how much and how many times?

The participant prefers to shop after work. The shopping only takes about 10 minutes and is done 3-4 times a week. furthermore the participant prefers to shop for a few days only, to keep

the food fresh. this results in a fridge that is quite empty, and as a result, the shopping is often a necessity.

What affects you in the shopping situation?

The participant is affected by "decorated" sales mend for tempting to do impulsive shopping, furthermore the participant does not like to buy the food that is on sale, because it is close to the best before date.

75 year old married female

- Sex: Female
- Age: 75
- Education: Vocational
- Work: Pensioner
- Household: 2, one married couple

What do you characterize as food waste, and is it a problem in the household?

No use of leftovers, and food that is badly stored, which makes it go bad. Food waste is not seen as a problem in the household, if there are small leftovers, the dog will get it.

Is there a difference in the type of food, that is thrown out?

The participant can not say what difference there could be, because food is never thrown out.

What factor is in focus, when food is thrown out?

If food would be thrown out, the freshness will be the only factor.

Planning meals and shopping.

The participant likes to shop large quantities of meat, because it is bought from a butcher who's shop is far away from home. Where items like bread and milk is bought as needed, the participant elaborates that there is rarely missing anything, because alternative ingredients will be found.

What is the most important about your diet?

It is important that the diet is healthy, and filled with energy, the meat has to be proper, and therefore it is bought from a butcher

Do you use leftovers?

The participant always use leftovers.

When do you shop, for how long, how much and how many times?

There is no specific time in the day which the participant likes to shop, and the shopping takes place 2 to 3 times a week.

What affects you in the shopping situation?

The participant likes to buy savings, and fill the freezer, and can be tempted to do impulsive shopping, furthermore the participant likes to buy the food that is on sale, because it is close to the best before date.

55 year old single woman

- Sex: Female
- Age: 55
- Education: Skilled Worker
- Work: Teaching assistant
- Household: 2, mother and daughter
- Phone: Yes, android
- Tablet: Yes, android

What do you characterize as food waste?

When food is thrown out in the trash. This applies to both prepared and unprepared food. If animals are fed with the food instead of it getting thrown out, it is not food waste.

Is foodwaste a problem for the household?

No, everything edible that gets too old, it is fed to the animals of the household, more specifically hens, as a replacement of fodder.

Is there a difference in what gets thrown out?

No, there is nothing specific that gets thrown out the most.

What factors are considered when throwing food out?

Mostly the freshness of the food. Not so much the best before date, but mostly how the food looks, feels, and smells.

When do you plan your meals and shopping?

The planning is done a 2-3 days ahead of when the food is getting eaten.

How is the shopping planned?

Mostly out of what is wanted to eat and discounts.

How is the coordination of the members of the household?

The mother makes everything, the daughter neither prepares the food or does any shopping.

What is valued the most about your diet?

What the household want to eat, variation of the meals, and a little economy.

Do you use leftovers?

Yes, leftover gets used for lunch or dinner, depending on the quantity of the leftovers.

When do you grocery shop?

In the morning before work, or after dinner.

How long does the grocery shopping take?

About an hour because there is being shopped in more than one store.

How many times a week is grocery shopping done?

3 times a week.

WHat is the quantity when shopping?

Depends from time to time.

What affects you in the groecry shopping situation?

The discounts are planned from home, as the participant looks through discount magazines. But if there is something practical and cheap, the participant will buy on impulse.

How do you make your shopping list?

The shopping list is made in by hand on a notepad, but if there was a simple solution for tablet or smartphone, the participant would be willing to try it out. The participant have already tried using Evernote for a shopping list that would sync between different devices.

28 year old single male

- Sex: Male
- Age: 28
- Education: Educator
- Work: Kindergarten helper
- Household: 11

What do you characterize as food waste, and is it a problem in the household?

The participant believes when eat able food is thrown out, is should be considered food waste. He does not feel that the food waste level in the household is a problem.

Is there a difference in the type of food, that is thrown out?

It is mostly vegetables, which is thrown out in the household, he does not always have the time to eat them all before they goes bad.

What factor is in focus, when food is thrown out?

When the participant feels the food has become to old he will throw it out, this is based on how fresh the food seems to be.

Planning meals and shopping.

How the participant plans his shopping and cooking varies, but the planning happens for the most part when he already is at the mall. When he is shopping for products, his choosing is based on what he likes and what there is on sale. He is alone when he is shopping or cooking.

What is the most important about your diet?

It can not be to spicy as he is vulnerable to stomach ulcers. It also has to be of a good quality and economic.

Do you use leftovers?

The participant tries to save and eat leftovers when possible.

When do you shop, for how long, how much and how many times?

He tries to do his shopping after work if possible. He is not sure about how much time he spends on shopping, but the time used goes down he know what he wants beforehand.

What affects you in the shopping situation?

He tries to get the wares he finds delicious, or if they are on sale.

B. PACT Analysis

B.0.2 PACT analysis

This appendix documents a PACT (People, Activities, Context & Technology) analysis performed for the project. The analysis is used during the understanding process in the Scenario-based design method.

People

In this analysis there will be focus on the social, physical and psychological differences between people and there will be a description of their different motives and preferences. The information used is gathered from the user stories in section 6.1.1, and from the information gathered in chapter 1, furthermore a description of this will be used to get an idea of who could benefit from a system, that would help them organize a household inventory, grocery shopping and preparing meals more efficiently.

Physical differences People have different physical abilities. Some people have bad vision, some have bad hearing, and so on, taking this into consideration is an important aspect of a PACT analysis.

There can be a great geographical distance between a person's home and used grocery stores. People who have trouble travelling this far, will need to go shopping as rare as possible, this can be helped with good planning and a detailed shopping list.

Looking at the ergonomically aspects of the program is not necessary, since the ergonomics are decided by the company who creates the device that the application run on.

Psychological differences People vary in the way that they function psychologically. When taking psychological differences into account, it will be most ideal to take the weakest user into account, so the program can be used by as many people as possible. Some common traits to look at when designing software are:

- The meaning of buttons
- Easy remembering of instructions
- Ease of use, through automated processes

The meaning of icons can vary, between cultures and countries, it is therefore very important to make sure that buttons will be perceived correctly. As a result form this, it cannot be certain that the program can be used in other countries than Denmark, which is where the program will be designed for.

It is also important that no instructions or commands are too long, because it will be hard to remember for some people, and when making a product you must take the weakest user into consideration when looking who to design the product for.

Automated processes reduce the amount of operations to remember, in order to do certain operations.

Social differences Social differences result in different requirements for a product. Below are mentioned different groups of people, whom have been divided into different categories.

People following a special diet, or having specific wishes for food plans, will have to buy food based on the diet, and sometimes prepare it in a specific way. Some examples of these people could be:

- Athletes (Bodybuilders)
- Vegans/vegetarians
- Organically minded

People who want to save time when planning, shopping, cooking and preparing the food, have to plan ahead and make shopping lists, to minimize the number of times they have to go shopping. Furthermore if they do not have time to shop all items on a shopping list, it should be possible to separate the most needed items from items, which can wait a set number of days, to be bought. Examples of people who can benefit from saving time because of tight schedules can be:

- Students
- Parents
- Families

People who want to be social while eating When people want to get together and eat for different occasions, it could be beneficial if they plan the meal based on the preferences of the people involved. The reason why people would like to be social while eating could be just wanting to talk to others, but also to save money by preparing bigger meals, and people trying to have less food waste by cooking together. Examples of people who would like to be social while eating can be:

- Social eaters, people who only get together to eat a meal together
- Students
- Parents
- People with a tight or small budget
- Students
- Parents

Comparison Looking at people, in different ways, different requirements can be set. Some of these requirements are that, different cultures and ways of understanding certain icons, must be taken into consideration, furthermore it seems to be beneficial to all, to receive help in planning meals, this can be done by shopping lists, and automated processes, to help generate this shopping list. Lastly, it would be beneficial to have help handling different cultural requirements, such as certain food products that should be avoided.

Activities

To see in which context the program will be used, we first look at the activities associated with using the program. The activities have been split into two categories, what happens *inside* and

outside of the home, furthermore the activities are based on the user stories section 6.1.1, and from the information gathered in chapter 1, as well as requirements fount in the people part, appendix B.0.2, of this analysis.

In the home:

- The program will be used to manage the food supply by:
 - Viewing a list of the stored groceries
 - Updating the list, by adding or removing groceries that could have been used, or groceries that have gone bad
 - Clearing the stored groceries. This should be done when the program is initiated for the first time or after a long period of inactivity
 - Create a shopping list of the groceries needed for the planned meals
 - Search for recipes
 - Add new or modified recipes
- Furthermore it should be in the home that preferences are set, for example groceries you want to ignore, because:
 - You do not like them
 - You are allergic
 - The grocery is not associated with a certain diet
- While cooking the meal, you would want to follow a recipe while cooking, which mean you might have to interact with the program during the cooking. Optimally a solution to this, would be to lessen the need of touching the devise, by having as much of the cooking information on the screen as possible, if interaction is needed, it should be done easily with as few clicks as possible, furthermore larger input areas will allow the user to f.x. use an elbow to navigate with.

Outside the home:

- While shopping for groceries, it is necessary to keep track of the shopping list, to see what should be bought.
- If a new recipe is wanted on the fly, the shopping list must be updated
- In certain circumstances it must be possible to define for how many days the shopping should be done.

Frequent activities such as looking for a recipe should be easy to do, but some activities such as clearing the list of stored groceries should be easy to learn. This could be done by having a walkthrough when a user tries to do this particular activity. The program should also have easy navigation options to allow for mistakes to happen, such as going into undesired program pages, and still be able to get back on track. If the user is interrupted and have to pause their usage of the program, the user should be able to continue later from the same point.

Searching for specific recipes can have a long response time, which could become a point of frustration for the user, especially if the search takes more than five seconds. As the recipes are available online there is a risk of high latency between a user's device and the server containing

the recipes. A workaround for this could be to have a local version of the recipe database, and only have the device synchronize the two databases once in a while.

Context

In this section we look at the context in which certain activities are performed. The examined contexts are the physical environment, and the social context.

Physical environment Using the program in different locations will have different impact on the user experience. Having the program on a device will set some limitations, as the user will have to interact with the program, during different situations.

- With the households inventory stored in the program, a user will not have to be home, to see what is missing, if he/she chooses to add another recipe.
- While shopping, one hand must be free, to check which items that has been put in the basket, and what else must be bought.
- While cooking at home, an ingredients list could be revisited, or a cooking guide will have to be followed, with greasy hands this kind of interaction will be difficult.

Social context The program can be used in different social context.

- If more than one person share the same food plan, they must both have access to the program, and if it is handled on their personal devices, synchronization is necessary.
- The program can be used by only one person and on one device only, therefore synchronization might be a nuisance instead of a trait for this type of user.
- With an online database of recipes, it would be possible to add new recipes and share these, with other users.

1

Technologies

In the technology part of the PACT analysis, there is looked upon input, output and communication, which will be examined by the pros and cons of different relevant technologies, with a focus on tablet and smartphone related technologies.

Input The program should function as a planning tool, and include a shopping list, therefore the final platform will likely be tablets or smartphones, since they are more mobile than a laptop, and therefore there might not be any external plug-in devices that are going to be used for the solution. This means that the available input technologies are:

- Cameras
- Microphones

¹Fixme Note: Conclude on this plz!

- Touchscreens (includes multi-touch)
- Physical buttons

The camera, can be used to input pictures, video or barcodes. Pictures and videos can be useful when the user adds content to the program, such as creating a recipe or adding a instructional videos. It is possible to get applications on some popular mobile operating systems such as iOS, Android and Windows Phone [**barcode_Phones**], that scan traditional barcodes and QR codes. Being able to read these codes could become useful if there are any relevant data on the 2D barcodes that needs to be transferred to the device. This data could be the item type or expiration date [**barcodeInc_FAQ**].

The microphone, can be used to input audio into the device. This can be useful if the users hands are occupied and they need to do some simple interaction with the program. These interactions would be made possible by the use of voice recognition software.

The touchscreen, is easy to use and can be found in tablets and smartphones. The user can press the screen with their finger to interact with the program. Some touchscreens also enables the use of multi-touch technology, this enables more than one finger to be used when interacting with the touchscreen, this is often used for operations such as zooming (pinching) and rotating of pictures and text. This is useful in the general use of the program.

Physical buttons, on the device could be used to navigate throughout the program, together with the standard functionality e.g. volume control. The amount of programmable buttons varies from device to device, the iPhone/iPad does not allow secondary functionality on their buttons. Some devices that run the Android OS allows for second functionality [**android_Rebind**], and can therefore be considered when designing the program.

Output The list of output technologies are:

- Screen
- Speakers
- Vibration

The screen, is used for visual display and visual feedback. Screen technology is useful to display large amount of information to the user. The size of the screen must be taken into consideration when displaying information such as screen elements (e.g. buttons) and/or text. Devices such as a laptop screen or desktop monitor can display many elements and large quantities of text, where as smartphones and tablets are restricted, both in resolution and dimensions. Large amounts of text can require pinching and scrolling making it hard to read. Also, large clutters or quantities of elements can be hard to navigate through. Lastly it can be useful for all screen sizes, to use icons to show what elements can do instead of using text, which can result in unnecessary use of screen space.

Speakers and/or vibration, can be used in conjunction with the screen to give different kinds of feedback simultaneously. Speakers and vibration are especially useful to give the user feedback in situations where the screen of the device is not visual. furthermore are the speakers helpful in situations were there are no physical contact with the device but it is still in hearing range. Vibration is useful when the user has physical contact with the device or in situations where sound might be hard to hear, or is disabled, this could be useful in crowded environments such as a shopping mall.

Communication, Communication covers how the solution is supposed to communicate with the internet and other users. The list of technologies used when using a smartphone or tablet, is:

- Wi-Fi
- Telephone networks
- Database

Wi-Fi, connection is a wireless connection to the internet. This is a useful technology because all modern tablets and smartphones has Wi-Fi technology. Wi-Fi can be used to synchronize data that is between devices, such as other smartphones or tablets, as well as an server

Telephone networks, is used for the same reason as Wi-Fi. In order to use this sort of network to connect to the internet, most mobile companies, require the user to pay for this access, furthermore, some tablets are cheaper when they only provide Wi-Fi technology, but it comes in handy, in the way that you can connect to it almost every where, where as with Wi-Fi, you have to be in range of a provider.

A *database*, can be used to store data on and the data can be accessed via the internet. This is useful because a database can store larger amounts of data than a mobile device. And it allows for easy access from multiple devises.

Conclusion Many different technologies are required to be considered when doing a PACT analysis, the technologies needed for the food planner application is from the three categories; input, output, and communication. The technologies ranges from physical buttons, to Wi-Fi needed for the synchronization of the data. The chosen input technology that is going to be used is the touchscreen, as it is already widely used and works well with the solution. The chosen output technology is going to be screen display with a possibility to include sound. The screen will show the necessary text and visual information, where the sound may be used for alerts or notifications. The communication technologies will going to be all of the previously described. Wi-Fi is useful for communicating with the server and other devices and the telephone networks can be used to transfer data when the user is outside of their home or work. The database will be used for holding a large amount of somewhat trivial data, such as the recipes. By having the recipes on the database the user saves room on their data storage device.

C. Personas

These personas are generated from user stories, in section 6.1.1 and conceptual scenarios, in section 6.1.2, furthermore the personas will be used as a helping tool for creating concrete scenarios and to help set requirements and a scenario corpus.

Henrik Jensen



- Age: 38
- Relational status: Single
- Occupation: Working as an IT consultant.
- Preferences: On a busy schedule Henrik rarely plan what to eat, until he on the way home drives by a store.

Henrik drives home after he has finished a meeting with a costumer. The meeting dragged on and Henrik just want to get home and get a nice dinner.

Henrik parks at the local store, where he starts the application. Thinking he want to cook something with chicken, he searches for recipes containing chicken, and finds 'Jamaican curry chicken', he add the recipe to his meal plan, and entering the store he navigates to the shopping list, containing the needed ingredients, which the application has not listed under storage.

Just as he enters the store he changes his mind on what to eat, because he sees some really cheap beef. He now searched for recipes with beef, and decides on 'swedish cocktail meatballs', next Henrik updates his meal plan thereby the shopping list has also changed.

When Henrik sees he has to buy eggs, he gets confused, because he is almost certain that he has eggs at home, but trusting the program he gets a pack of eggs.

When Henrik is home and begins to unpack, he notices that he actually did not have any eggs, and is therefore happy he trusted the program.

Peter Nielsen



- Age: 23
- Relational status: With girlfriend
- Children: None
- Occupation: Student
- Preferences: Does not spend more time in the kitchen than needed.

Sitting in class, Peter knows that he should be the one to buy what is needed for dinner, but Peter is not the most imaginative when it comes to deciding this, so he texts his girlfriend, who at the time is not home either, she tells Peter to just get what is needed for the recipe, where they already have the most ingredients. Peter looks up a list sorted by what they already have, with the most completed lists at the top, he finds 'Beef, Snap Pea and Asparagus Stir-Fry', and adds it to the meal plan for the evening

On his way home Peter goes to the closest grocery store, and shops the needed ingredients, and without neither him or his girlfriend needing to be home to see what was needed to be bought, they had what they needed to prepare the meal.

Anne Madsen



- Age: 18
- Relational status: Single
- Children: None
- Occupation: Student
- Preferences: Likes to spend time in the kitchen to prepare a good fresh meal, as well as spending time on preparing what meals to have, days in advance.

Anne has the day off, from school, when she looks at her meal plan, to find that she has only planned for the next two days. Since she has the time she decides to plan the next two full weeks.

Using the application Anne quickly fills out all the days, and with more time to spend, she wants to do the shopping for the next four days. She therefore sets her shopping list to only include the next four days, then she goes to the city, to do the shopping.

While doing the shopping she notices that none of the stores she has been to, have two of the ingredients she needs for the meal on the fourth day. Knowing that she has to go to another part of the city the day after tomorrow, she sets the shopping list to only show three days, and then continues the shopping, only getting a few of the items needed for the fourth day.

D. Design requirements

This list shows requirements used in section 6.1, the requirements are listed into different domains and a more general list, followed by a list of requirement actions and a limited list of the technologies which are available.

Planning

- It must be possible to use leftovers.
- Base recommended recipes on what is already at home, lessening the need of shopping.
- Must be possible to plan for a different numbers of days.
- Being more than one person to plan, and especially if the planning is not detailed enough, it is easier to buy wrong items, or to forget to buy items, but it can help to improve the time it takes to shop, and lessen the amount of impulsive shopping. Furthermore, by having a long time to plan, there will also be more time to look through recipes and compare with the inventory.
- Certain people have to or chose to follow specific diets. Taking this into consideration, will expand the target audience furthermore.
- Another aspect that can be taken into consideration, is that people for different reasons could want to meet with other people. Therefore a feature to help gather people for social meetings, could be a possibility.

Shopping

- Users will need to be able to plan meals, while they are out shopping.
- Some users are more willing than others, to go to multiple stores while shopping.
- It varies when in the day, and how many times in a week people are willing to shop.
- Planned shopping lists reduces the shopping time for some users.
- Different aspects, while shopping can change what users what to buy, these aspects can be sales, impulsive shopping, or finding some interesting items that are on sale because they are close to the best before date.
- Not having a shopping list tends to result in buying to much, because it is not sure what should be bought, also people tend to do more impulse shopping, and having to go for more trips.
- Unfamiliar stores, where the user is not sure what can be bought in the specific shops, as well as long shopping list can result in a greater risk of having to visit more stores.
- Depending on the distance between home and stores, different requirements can be set, because a long distance can result in having to shop bigger quantities and it is more important not to forget to buy items.
- While shopping, one hand must be free to check the shopping list.

Cooking

- Cooking can result in greasy hands, and if it is necessary to navigate in the program at the same time, it would have to be easy to do this navigation.

Inventory

- with the inventory list in the program, it can be followed on the fly, and thereby the shopping list can be updated when the meal plan is changed.

General

- Quality, organic, exiting, healthy and varied meals are important factors to users.
- Buttons in a program can be interpret in different ways, it is therefore important to make sure that the right icons are used.
- The more simple instructions are to follow, the wider a target audience can be.
- By having automated processes, the program will become easier to use, and thereby also widen the target audience.
- If the same storage is to be used on multiple devises, it would require synchronization between the devises.

In the following there are listed actions/requirements which happen in and out out the home, they are not sorted into domains because they are based on actions, actions/requirements to handle are:

- View list of stored items
- Update the stored item list, with used ingredients or ingredients that get old.
- clearing the whole stored items list. Could be that many items go bad, during to a holiday.
- Create a list of meals, that can be set to specific dates.
- Search through a list of recipes.
- Add or modify recipes.
- A list of items to ignore.
- While shopping, it should be possible to trace the status of the shopping list while shopping.
- If the meal plan is changed, the shopping list should change accordingly.
- A 30 day meal plan should not need to be bought the same day, therefore it should be possible to define the number of days to shop for.

The program will also have to use different technologies, these are described in the technology part of the PACT analysis, appendix B.0.2 are as follows;

- Touch screen
- Camera
- Microphone
- speakers
- Wi-Fi

- Telephone network
- Database

E. Usability Test Tasks

E.0.3 Usability Test Tasks

When the test subjects of the usability test had been told how the test was conducted, they were given a list of tasks that they had to perform within the program, to test the usability. These tasks are described underneath.

Task 1 You have nothing planned for tonight's meal, and you feel like having *chicken and leek pie*.

- Search for and add the *chicken and leek pie* recipe for today's meal

Task 2 On the way home from work, you want to do some shopping for tonight's meal, but because you are on a bike, you cannot shop many items; therefore you want to change the number of days to shop for.

- Navigate to the page that allows you to change number of days to shop for, and change this to 1.

Task 3 You have been doing some shopping for tonight's *chicken and leek pie* and want to add some bought items to your inventory

- Find *onion*, *egg* and *leek* on the shopping list and add them to the inventory list.

Task 4 You had a dinner planned with some friends, but due to bad planning from your friends, you have to cancel your plans.

- Go to week 51, and remove *Spinach and Cheese Souffle* from Monday the 15th of December.

Task 5 During the last shopping trip you also bought 5 *lollipop sticks*. These have an expiration date that says *16th of July 2017*.

- Add 5 lollipop sticks to the inventory and change their expiration date to *16th of July 2017*

Task 6 Friends have invited you over for a party Sunday the *14th of December*, so you have to reschedule the meal you have planned for this day

- Change the date of the meal on the *14th of December* to be scheduled for a week later.

Task 7 After a long day, you feel like being good to yourself, you therefore eat some of the *white chocolate* you have.

- Remove *100 ml of white chocolate* from the inventory

Task 8 You have some friends over, when you decide to have some *beer*

- Remove all *beer* from the inventory

Task 9 It is Christmas and you feel like having something with *cinnamon* in it.

- Search for recipes with *cinnamon* and add one for today.

Task 10 You are having friends over for dinner *Friday the 19th of December*; you know that you will be 9 persons in total

- Find the recipe *Hot and Smoky Cheeseburgers with Bacon and Pickled Cherry Pepper Relish* and add it for the *19th of December*
- Change the number of meal participants to 9

Task 11 Because it is Christmas you decide that you want more recipes with *cinnamon* to be shown.

- Add *cinnamon* to the list of rated ingredients with a rating of 100

Follow-up questions

After a test had been conducted, some follow-up questions were asked to find if the test participants had any comments about the usability of the program.

- Do you have any general comments about the program?
- Did you find the program easy to use?
- Was the text easy to read, even though some of it were orange, and the background where grey?
- Did it make sense that the functionalities in the settings screen where placed there, or should they have been placed in other pages?
- Did the rate ingredients function make sense? Was 100 too many? Would 10 stars have been better?
- How did navigation work through the program? Was it easy to navigate through pages?

List of Corrections

Note: skriv mere præcist nå testing er done	3
Note: beskriv bedre når det er lavet	3
Note: references to bibliography, and/or maybe some graphs to show the numbers?	5
Note: possibly to repetitive	8
Note: textbf might be a bad choice.. CHS tried using emphasize on Less gro... and subsubsec on Modules (subsubsec throws a warning because we have no subsec beforehand .. please evaluate, on which type we should use.	9
Note: Brug slides fra lektion 3 SU for at skrive introer til afsnit.	12
Note: resize elementer så de er lige store + kosmetiske fejl	12
Note: Figure 2.1 can also be divided into different context areas. The areas represent contexts in which different symbols and processes are performed most often. The contexts are in the home and out of the home.	14
Note: CHS: Still don't know how or if, to incorporate metaphors	14
Note: "Should I add more pics of Food Planner app?" Christian asked the group.	16
Note: what is this (explain in definition below)	17
Note: Skriv intro	58
Note: cut some of the top of this fig. + maybe put numbers 1 to 5 on the colours, and refer to these, and not only the hashcode	62
Note: Skriv om DietRule og DietPreset	67
Note: explain why similar classes does not inherit or relate	67
Note: this is not fully written.	68
Note: skriv?	72
Note: explain	72
Note: explain	72
Note: Describe sorting	73
Note: Describe this	75
Note: Conclude on this plz!	98