

P1 - Et program der løser et problem

Gruppe 3, SW1 *

December 2021

*Anders Mazen Youssef, Rasmus Bunkenborg, Silke Delfs Nedergaard,
Simon Alexander Larsen, Sofia Gran, Thor Koustrup

Abstract

This paper explores the problem of food waste and discusses why young adults in the age group 18 - 34 years are the group that produces the most food waste in Denmark.

Various forms of food waste are defined and explained, and statistics document the extent of the problem are presented. By analysing studies examining the eating habits of young adults we get a better understanding of the problem, and why young adults are over-represented in food waste statistics. The aim is to create a program that will contribute to the reduction of food waste by helping young adults to prepare dishes based on the ingredients they have available in the fridge.

The program is made in C and consists of two main functions, a digital fridge and a recipe generator. The digital fridge constantly shows what ingredients you have available, and the recipe generator generates recipes based on these ingredients. By having recipes that are easy to make, and are based on what you have available, it can motivate young adults to reduce their food waste, and at the same time reduce their food expenses. Our program is in many ways just a prototype and an inspiration for future work. For a more sustainable future, and to further reduce food waste, we encourage further development of the ideas and functions we ourselves did not have time to implement.

Forord

Denne rapport er skrevet i forbindelse med projektet “Et program der løser et problem” som led af første semester på bacheloruddannelsen Software på Aalborg Universitet København. Rapporten er skrevet af gruppe 3. Rapporten blev skrevet over perioden 11. oktober 2021 til den 16. december 2021. Projektperioden involverede intensiv gruppearbejde og fælles programmering.

Rapporten er skrevet med det formål at skabe et dybere indblik i konsekvenserne af unge voksnes madvaner, særligt målgruppen 18 til 34 årige. Fokus for denne rapport er at dokumentere og undersøge hvorfor lige netop denne målgruppe er ansvarlig for den største andel af madspild i de danske husholdninger. I den forbindelse redegøres for konsekvenserne af madspild, og hvordan man potentielt kunne afhjælpe disse vha. at skabe et værktøj i form af et program, der kan hjælpe målgruppen med at skabe større struktur i over deres spisevaner og overblik over deres madspild. I vores projekt skrives program-løsningen i C.

Tak til vejleder Henning Olsen og dem der har valgt at svare på spørgeskema, samt dem der valgte at fungere som testpersoner.

Indholdsfortegnelse

1	Indledning	1
1.1	Madspild - definition og data	1
1.2	Unge madvaner	4
1.3	Afgrænsning og problemformulering	6
2	Metoder	8
2.1	Teori/pensum og empirisk grundlag	8
2.2	Research	9
2.3	Egen forskning	9
2.4	Brugertests	10
2.5	Samarbejdsmetoder	10
2.6	Begrundelse for fravalg	11
2.7	Kritik af resultater	11
3	State of the art	13
3.1	Allerede-eksisterende løsninger	13
3.2	Mad for fattigrøve af Fie	13
3.3	Sulten	14
3.4	Too Good To Go	15
3.5	Ligheder og forskelle mellem eksisterende løsninger	16
4	Analyse	18
4.1	Spørgeundersøgelse	18
4.2	Konklusion af Spørgeundersøgelse	23
4.3	Målgruppeanalyse	24
4.4	Den nye problemformulering	26
4.5	Det potentielle program	27
4.6	Persona	28
4.7	Scenarier, en fiktiv test med persona	29
4.8	Kravspecifikationer	36
4.9	Prioritering af krav (MosCow)	38
5	Design og implementering	41
5.1	Flowchart over design	41

5.2	Implementering	43
5.3	Testtyper	49
5.4	Brugertest	52
6	Diskussion, perspektivering og konklusion	55
6.1	Diskussion af programmet	55
6.2	Perspektivering	56
6.3	Konklusion	57
7	Evaluering	58
	Bibliografi	59

1 Indledning

1.1 Madspild - definition og data

Hvert år bliver 814.000 tons spiseligt mad smidt ud i Danmark[1]. Det viser tal for hele fødevareforsyningskæden fra Miljøstyrelsen i 2021, og de helt store syndere viser sig at være henholdsvis fødevareindustrien med 385.000 ton og almene husholdninger med 247.000 ton årligt[1]. Dette svarer til, at i gennemsnit lader den almene dansker ca. 42 kg mad gå til spilde om året[2].

For at forstå hvad madspild er og ikke er, skal vi først afklare hvad begrebet dækker over.

Madspild bliver defineret af Miljøstyrelsen som “fødevarer, der kunne være blevet spist af mennesker, men som går til spilde”[3], og undersøgelser viser at ca. halvdelen af al den mad, der bliver smidt ud i de danske husholdninger, kunne være blevet spist[4]. Det er således langt fra kun mad, der er udløbet eller af anden årsag ikke kan spises, som jævnligt bliver smidt ud. Det er vigtigt at påpege, at madspild i denne definition dermed også er f.eks. rester, der bliver fodret til kæledyr, eller grøntsager der ryger til kompostbunker og lignende.

Derudover er madspild en underkategori af madaffald, der inddeles i to kategorier - førnævnte definition af madspild, og øvrigt madaffald, der ikke er beregnet til at blive indtaget af mennesker såsom eksempelvis kaffegrums og æggeskaller[5].

I denne rapport arbejdes der udelukkende med madspild, som altså kunne være indtaget af mennesker. Senere i rapporten indskrænkes dette emne yderligere til kun at omhandle madspild i de danske husstande, og igen til madspild i en bestemt befolkningsgruppe - navnlig de unge voksne i målgruppen 20 - 34 årige.

Det bør påpeges, at madspild i Danmark sker på alle niveauer i fødevareforsyningskæden, der betegner de forskellige aktører i fødevareproduktion - lige fra landbruget til spisebordet. Virksomheder har stor indflydelse på madspild, og der bliver

selvsagt bidraget til madspild under dyrkning, produktion, transport, håndtering og indpakning i butikkerne, samt tilberedning og servering i restauranter. Det er tydeligt at se, at der går en del ressourcer tabt fra produkternes vej fra den primære produktion til forbrugernes tallerken og videre endda - også endda når det specifikt handler om madspild, og altså ikke blot madaffald i den brede definition. Det er således ca. 567.000 tons mad, der går til spilde eller på anden vis gøres ubrugeligt, inden det overhovedet når hjem på tallerkenen i husstandene.

Værdikædeled	Vurdering af madspild i ton pr. år, data fra Miljøstyrelsen 2021
Primærproduktion	44.000
Fødevareindustri	385.000
Detail og engros	96.000
Restaurationsbranchen	42.000
Husholdninger	247.000
I alt	814.000

Figure 1: Mængderne af dansk madspild fra Miljøstyrelsen - 2021

Vores fokus vil imidlertid være den almene dansker, og hvordan man kan forhindre madspild blandt dem. I og med at de danske husstande står for ca. 30 % alene af madspildet i Danmark per 2021, er det nærliggende at undersøge hvordan man kan mindske dette.

Privat madspild er både et økonomisk problem og et miljøproblem. Det er selvsagt dyrt at spilde mad, både for pengepungen og for klimaet. Den gennemsnitlige danskers forbrug udleder ca. 17 ton drivhusgas om året, og en artikel skrevet af den grønne tænketank CONCITO[6] anslår at omkring 3 af disse 17 ton er forbundet med vores fødevareforbrug. I den forbindelse er det værd at nævne, at der kan være meget stor forskel på hvor stort et aftryk produktionen - og dermed spildet - af en bestemt type madvare har i forhold til CO₂-udledning.

En rapport fra EU i 2015 beskriver i et case-study: “The fruit and veg-

etable department was responsible for 85% of the total waste, but only 46% of the total carbon footprint. The meat department, on the other hand, was responsible for only 3.5% of total waste, but 29% of the total carbon footprint[7].” Det kan derfor være vanskeligt f.eks. at anslå hvad ét kilo madspild i den almene husstand svarer til i kilo CO₂-udledning[8]. Firmaet bag TooGoodToGo-appen, som er en app der sælger ud af overskydende fødevarer fra detailhandlen, anslår dog at i gennemsnit svarer 1 kg madspild fra en almindelig husstand svarer til 2.5 kg CO₂[8].

FN har 17 Verdensmål for Bæredygtig Udvikling, der inddeles i 169 delmål. Mål nummer 12 hedder Ansvarligt forbrug og produktion, og herunder finder man delmål 12.3 der lyder:

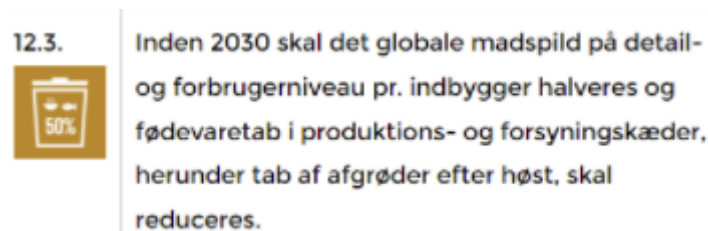


Figure 2: FNs Delmål 12.3 fra FN

Her er hovedfokus altså at reducere tabet af fødevareressourcer i fødevareforsyningskæden, sådan at produktion af fødevarer foregår på en mere bæredygtig måde og med et langt mindre CO₂-aftryk. FN anslår, at på globalt bliver over en tredjedel af alle fødevarer spildt eller smidt ud - svarende til 1.3 milliarder tons, eller madspild for rundt regnet 1 trilliard USD[9].

Et af indsatsområderne, der skal sættes ind på for at nå dette mål, er at ændre på den måde folk i 1. verdenslande forbruger sine fødevarer, da problemet med for stort madspild i de individuelle husstande er et stort problem i disse lande[9]. Det skal dog påpeges, at der som tidligere nævnt sker et endnu større spild af fødevarer længe før forbrugerne får deres indkøb med hjem. Der er absolut et behov for effektivisering af produktion og behandling af fødevarer på alle niveauer i fødevareforsyningskæderne på globalt plan, ligesom i Danmark, hvis man skal mindske det miljømæssige pres på naturen. Det står klart, at det enorme madspild på verdensplan bidrager betydeligt til klimaforandringerne. Fødevarer, der produceres men aldrig indtages af mennesker, er alene ansvarlig for over 6 % af udledningerne af drivhusgasser

på verdensplan[10]. Til sammenligning udleder lande som Kina og USA henholdsvis årligt 21 % og 13 % af al drivhusgas på verdensplan[10].



Figure 3: Mad produktion og co2 udledning, fra Our World In Data

Imidlertid er der en del man kan gøre som individ, og det handler ikke mindst om at ændre sine indkøbsvaner. I den tidligere nævnte artikel fra CONCITO nævnes der nogle generelle gode råd til at handle ind på en ansvarlig og bæredygtig måde:

“Køb kun den mad, du har brug for, tilbered kun den mad du kan spise og gem resterne enten i køleskab eller fryser, så de kan spises en anden dag. Kombinerer du dette med en klimavenlig kost med grønne sæsonvarer og et begrænset kødforbrug, er du godt på vej til at minimere din fødevarerrelaterede klimabelastning og mindske unødigt madspild.”

Og det er også det fokus, vi vælger at arbejde med i denne rapport.

1.2 Unges madvaner

I en undersøgelse fra 2018, som Fødevarestyrelsen bestilte fra DCA – Nationalt Center for Fødevarer og Jordbrug - undersøgtes danskernes forståelse og opfattelse om madspild, samt de vaner og psykologiske faktorer der har sammenhæng med madspild[11]. Formålet var at finde mulige løsninger på at reducere madspild i de danske husstande. I Danmark er der generelt stor bevidsthed om madspild og bæredygtighed, men der er dog stadig er mulighed for at forbedre danskernes forståelse i forhold til konsekvenserne ved madspild. De danske forbrugere misforstår også i mange tilfælde hvordan madspild betegnes, da mange f.eks. ikke er klare over den førnævnte definition af hvad madspild er - nemlig mad, der ikke bliver indtaget af mennesker.

Størstedelen af dem, der deltog i undersøgelsen lavet af DCA mente selv, at de smed meget lidt mad ud. Dog var der også en stor del der gemte madrester, uden at vide om de ville blive spist eller ej. Da deltagerne blev spurgt om hvilke incitamenter, der ville motivere dem til at reducere madspild, så var besparelse af penge det de fandt vigtigst - ikke umiddelbart bæredygtighed.

I samme undersøgelse fandt man, at unge i alderen 18-34 år er den aldersgruppe, der smider mest mad ud, og det er der nogle særlige grunde til. For at kunne forstå hvorfor unge spilder så meget mad ud, er man først og fremmest nødt til at sætte sig ind i den hverdag langt de fleste unge voksne befinder sig i.

Det er et faktum at unge voksne i dag føler et enormt pres og stress. Undersøgelser af Dansk Center for Undervisningsmiljø fra 2012 viste at hver femte elev på ungdomsuddannelserne føler sig stresset "det meste af tiden", og hver fjerde af Dansk Magisterforenings studentermedlemmer har haft alvorlig stress på det seneste semester[12]. En undersøgelse fra 2018 af EVA - Danmarks Evalueringsinstitut - viste at dette ikke er blevet bedre med tiden, og at "hver tredje nye studerende føler sig meget stresset"[13]. De studerende som EVA spurgte ad fremhæver, at faglige krav, tidspres og personlige forhold er hovedårsagerne.

Med udgangspunkt i disse undersøgelser, er det meget tydeligt at en travl hverdag med lektier, afleveringer, eksamener, studiejobs, udover et pres om at have et aktivt socialt liv, fører til at mange unge voksne er stressede. De føler at de ikke har nok tid til at hverdagen går op, og derfor bliver de enkelte praktiske gøremål nedprioriteret. Her er det især madplanlægning, madlavning og at bibeholde de gode kostvaner, der forsømmes.

Overskuddet til at planlægge madlavning er meget lille, og det kan være svært at holde en ugentlig madplan - hvilket jo fremhæves som et middel til at mindske madspild i rapporten fra DCA. Et andet problem, der gør det svært at handle ind på en bæredygtig måde, er at mange fødevarer kun kan købes i store mængder og multipakker. Det er en udfordring at købe ingredienser nok til kun én enkelt opskrift, specielt hvis man skal købe ind til kun en eller to portioner. Dette fører til mængdekøb, som fører til ekstra madvarer i køleskabet, som desværre ikke altid når at blive brugt pga. de unge voksnes travle og uforudsigelige hverdag. Af samme årsag spædes der ofte til de uplanlagte måltider i hverdagen med impuls køb og take away.

Udover at man let kan miste overblikket over hvad man har er ingredienser, er manglen på kundskab inden for madlavning og brug af rester også et problem der leder til madspild. En anden rapport skrevet af Madkulturen, som er en selvejende videns- og forandrings-organisation under Ministeriet for Fødevarer, Landbrug og Fiskeri, undersøgte specifikt årsagerne til unges madvaner og -spild. Hovedkonklusionen fra Madkulturens rapport lyder:

“En uforudsigelig hverdag er kilde til unges madspild, økonomien vægtes over klimaet ved køledisken, og de unge mangler kompetencer til at bruge det, de allerede har i køkkenet” [14].

Hvilket stemmer overens med rapporten fra DCA. Det er derfor en dokumenteret problemstilling blandt de unge voksne, at man lige netop i denne målgruppe begår et unødvendigt stort økonomisk og fødevaremæssigt spild, der bidrager til et tab i bæredygtigheden indenfor fødevareforsyningskæden i Danmark.

1.3 Afgrænsning og problemformulering

I dette projekt er vores opgave at skabe et program, der løser et problem. Vi har valgt at tage udgangspunkt i emnet “Digital Madplan”, og fokusere på målgruppen unge voksne i alderen 18 - 34 årige. Eftersom målgruppen både er den, der producerer mest madspild i Danmark, og vi selv er en del af denne, finder vi det stærkt relevant at forsøge at komme med løsningsforslag til de problemer der ligger til grund for den uheldige udvikling.

Som beskrevet i indledningen er klimaforandringerne et stort problem, og vi ønsker at bidrage til en bæredygtig udvikling ved at påvirke unge voksne til at træffe mere bæredygtige valg. I denne kontekst er vores fokus at mindske denne målgruppes madspild, som har en global og direkte effekt på Jordens ressourcer og klima. Et reduceret madspild vil på sigt bidrage til mindre udledninger af drivhusgasser, da produktion og forbrug retter ind herefter. Samtidigt er der en økonomisk gevinst i at tilrettelægge de unge voksnes forbrug på en sådan måde, at de ikke utilsigtet køber for meget eller for dyrt ind pga. en travl hverdag.

Det formodes, at målgruppen kan motiveres på dette grundlag, da økonomi

og billigt indkøb som tidligere nævnt i nogen grad vægter højere end bæredygtighed for de unge voksne. Et andet fokus bør derfor være at skabe et program, der gør at reducere af eget madspild hverken er tidskrævende eller svært, men tværtimod let og en økonomisk fordel.

Der er altså to hovedårsager til de unge voksnes store madspild, og de er: at de unge voksne har for travlt en hverdag og for lidt tid til at planlægge madindkøb. Et mistet overblik fører til uheldige indkøbsvaner, som fører til et stort spild både økonomisk og miljømæssigt. at de unge voksnes manglende kundskaber i køkkenet gør, at de ikke formår at bruge hverken de ingredienser de sidder inde med eller de rester de producerer.

Dette leder os til vores problemformulering:

Hvordan kan vi lave et program, der reducerer madspild i målgruppen af unge voksne mellem 18-34 år?

2 Metoder

2.1 Teori/pensum og empirisk grundlag

Denne rapport er baseret på de erfaringer og den viden, vi har fået under kurserne Imperativ Programmering (IMPR), Datalogiens Teoretiske Grundlag (DTG) og Problembaseret Læring (PBL). Programmet laves i C, da pensum på nuværende tidspunkt kun har dækket dette ene programmeringssprog i kurset Imperativ Programmering. De emner fra vores kursus, der er mest relevante for vores program, er kapitlerne om udviklingsmiljø og kompilering, imperative principper, datatyper og variable, kontrolstrukturer, funktioner og procedurer, datastrukturer herunder arrays og sammensatte datastrukturer.

I en artikel, publiceret for et magasin der skal hjælpe både erfarne og nyere programmører, står det skrevet at C er det bedste valg når man skal lære at programmere for første gang.[15] En af grundene til dette er, at C ikke har lige så mange indbyggede metoder og biblioteker som andre programmeringssprog. I C må man håndtere software-relaterede problemer, som mange andre moderne programmeringssprog gør automatisk for brugeren. På den måde bliver man bedre til at løse problemer i C, da man selv er nødt til at implementere de forskellige funktioner der kan være brug for. Dette giver også en bedre forståelse for hvordan software fungerer. Programmeringssprog med mange pre-installerede funktioner og indbyggede biblioteker gør, at man ikke kan se al den underliggende kode der ligger bag de automatiserede funktioner. Dette fænomenet er kendt som “black-box”. C handler i stor grad om at man må lave alt selv fra bunden af, og det giver en helt unik forståelse af programmering.

Når man lærer at programmere i C, er man derfor nødt til at gå systematisk og overskueligt til værks, når man skal skrive den nødvendige kode. Denne systematiske tilgang til kodning kan med fordel hjælpes på vej af trinvis forfinelse, top-down programmering og “divide and conquer”-metoden, og det er også disse metoder vi i dette projekt har arbejdet med. C kræver altså mere af dig som programmør, men kundskaben og erfaringen du får af i gennem implementering af kode i C kan siges at være god at have med videre, når man senere skal arbejde med andre programmeringssprog.

Når det kommer til planlægning og selve skriveprocessen, er det lagt fokus

på den viden, vi har fået gennem kurset Problembaseret Læring og bogen "Problembaseret læring og projektarbejde ved de videregående uddannelser". De kapitler, der omhandler udvikling af problemformulering, planlægning og samarbejde har haft stor betydning i projektets opstartsfasen. Derudover har lektionerne om videnskabsteori og videnskabelige metoder været nyttige når vi har arbejdet med analyse og research. En vigtig del af den videnskabelige metode, især i arbejde med en videnskabelig rapport, handler om, at stille sig tre videnskabsteoretiske spørgsmål.

Det ontologiske spørgsmål: "Hvilke dele af virkeligheden skal jeg undersøge for, sådan at jeg får den viden, jeg ønsker?"

Det epistemologiske spørgsmål "Hvordan kan jeg få den viden, jeg vil finde?"

og subjekt/objekt-modstillingen eller forholdet mellem analytisk og empirisk objekt: "Hvordan er forholdet mellem mig, der undersøger, og det, jeg undersøger?". [16]

Disse tre spørgsmål blev aktivt brugt gennem hele processen.

2.2 Research

Projektet startede med en omfattende research, hvor målet var at få et overblik over problemfeltet. Da den endelige problemformulering var fastlagt, gik vi over til en dybere research med fokus på vores målgruppe og deres madvaner. Her brugte vi omfattende undersøgelser fra kvalificerede aktører som blandt andet Fødevarestyrelsen, Miljøstyrelsen, DCA - nationalt Center for Fødevarer og Jordbrug, og Madkulturen - en selvejende videns- og forandringsorganisation under Ministeriet for Fødevarer, Landbrug og Fiskeri. Disse undersøgelser har bidraget til at forstå unges voksne madvaner og behov, samt lagt et fundament for videre forskning.

2.3 Egen forskning

På baggrund af disse undersøgelser lavede vi et kvalitativt spørgeskema, som blev sendt til vores medstuderende på Software 1. semester ved AAU København. Spørgeskemaet gik i dybden med elevernes madvaner og behov, og målet var at bekræfte/afkræfte de hypoteser vi havde på forhånd,

samt føre os frem til mulige kravspecifikationer på baggrund af resultaterne. Spørgeskemaet bekræftede sammen med de andre undersøgelserne mange af de hypoteser vi havde, og gav os et godt grundlag for at definere hvilke funktioner der er vigtige at vores program indeholder.

Spørgeskemaet er sammen med de undersøgelser lavet af de nævnte aktører grundlaget for vores målgruppeanalyse, persona og scenarier. Persona og scenarier blev brugt som en metode for at teste en tænkt prototype af vores program, og finde eventuelle mangler. Scenarier gav os mulighed for at teste “programmet”, selvom programmet ikke var skrevet færdigt. Vi havde en klar idé om hvordan programmet skulle fungere, og ved at opsætte scenarier baseret på denne idé, var vi i stand til at hurtig redegøre for og reflektere over mulige fejl og mangler ved programmet.

2.4 Brugertests

Vi har taget udgangspunkt i “user-centric service model”. Denne går ud på at teste programmet, for at så gennemføre en evaluering om fører til en ny kravspecificering, der igen fører til tests og evaluering. Vi gennemførte en brugertests af programmet, hvor to personer sammen testede programmet og kom med feedback. Testen blev gennemført på skolen, og testpersonerne fik derfor ikke mulighed for at prøve opskrifterne. Målet med testen var at teste brugervenligheden og finde eventuelle problemer, der kunne opstå ved brug af programmet. Der blev lavet et kort interview efter testen, hvor testpersonerne kom med feedback på programmet. Feedback fra testpersonerne blev brugt til at rette op små bugs og lave en bedre opstilling. På grund af tidsmangel var det ikke muligt at lave hverken flere funktioner i programmet, eller mere uddybende tests.

2.5 Samarbejdsmetoder

Et godt samarbejde var vigtigt, da vi som gruppe skulle om at lave et fælles program og en fælles rapport. Vi benyttede GitHub med det formål at kunne arbejde på vores program sammen, men på hver vores computer. Vi oprettede et “repository” hvor vi kunne udgive vores code i et fælles online drev, hvilket også gav os mulighed for en “historie” side, der giver et klart overblik over hvilke ændringer der er blevet lavet til programmet. Derudover, så

benyttede vi også Microsoft Teams. Denne kommunikationsplatform blev primært brugt til at arrangere møder med vores vejledere, samt gruppearbejde hvis vi valgte at arbejde hjemmefra. Messenger var vores primære kommunikationsplatform, her delte vi ofte links og arrangerede tidspunkter hvor vi skulle mødes. Google Drive benyttede vi til at have alle vores dokumenter i et samlet online drev. Dette gjorde det muligt, at have en samlet oversigt over alle gruppemedlemmernes dokumenter, samt at kunne skrive kommentarer til hinanden. I bilaget "Procesanalyse - P1 SW1 2021" er det skrevet mere detaljeret om samarbejdsmetoder.

2.6 Begrundelse for fravalg

Da vi først havde et test-klart program i den sidste uge af projektet, havde vi ikke tid til at gennemføre de test, som vi havde planlagt på forhånd. Vi havde et ønske om at have flere testpersoner, som kunne teste en prototype af programmet over en periode på en uge, efterfulgt af et interview. I interviewet ønskede vi at få information og feedback på, hvad der fungerede godt, og hvad der ikke fungerede godt. Baseret på disse oplysninger, kunne vi have foretaget ændringer og forbedret brugervenlighed. Målet var at køre flere test eftersom vi redigerte/forbedret programmet, men dette fik vi ikke tid til.

Et andet alternativ vi valgte væk, primært på grund af tidsprioritering, var at have dybdegående interviews med eksperter og brugere før vi lavet en kravspecifikation. Dette kunne potentielt været med på at få viden om, hvilke krav vi skal stille til vores program, med henblik på at løse vores problem. Til gengæld gav de undersøgelser vi har kigget på, sammen med vores eget spørgeskema og scenerier, et godt udgangspunkt for en kravspecifikation.

2.7 Kritik af resultater

Programmering i C har både fordele og ulemper, og det uddybes i kapitlet om design og implementering. I C er det ikke muligt at lave et program der kan deklarere nye variabler baseret på input fra brugeren. Det betyder, at brugeren ikke kan selv oprette nye ingredienser til køleskabet, som ikke allerede er deklareret. Derfor skal du på forhånd have lavet en samlet liste, hvor du har deklareret alle mulige ingredienser. Her har vi begrænset os til

de nødvendige ingredienser til opskrifterne, og denne liste kan opdateres i takt med at opskrifter bliver langt ind. En anden vigtig begrænsning i C er, at man ikke kan lave en app eller en hjemmeside, eller for den sags skyld skrive en brugervenlig user interface. En potentiel prototype vil derfor også være begrænset af dette, da det ikke er muligt f.eks. at få programmet ned på en smartphone til behændig test ude i det virkelige liv.

Vores spørgeskema blev kun sendt til vores medstuderende, og vi modtog i alt 28 besvarelser. Selvom spørgeskemaet har givet os viden, der har stor relevans for udviklingen af vores program, har det nogle svagheder. For at få et endnu mere pålideligt resultat, kunne vi sende spørgeskemaet til et bredere udvalg af unge voksne, og dermed få et mere sikkert resultat. Derudover kunne vi, som nævnt i sidste afsnit, have gennemført dybdegående interviews der validerer og giver de påstande vi fremsætter, større troværdighed. Til gengæld henviser vi til flere kvalitative undersøgelser af høj kvalitet og troværdighed, som bygger troværdigheden af de påstande vi fremsætter.

3 State of the art

”State-of-the-art” handler om at finde den nyeste teknologi og information indenfor det felt man, arbejder med. I forhold til problemformuleringen vil denne teknologi og empiri være knyttet til reduktion af madspild og forskellige former for madplaner, samtidig med at vi må tage forbehold for at vi på 1. semester kun kan programmere i C. Dette kapitel vil derfor handle om at præsentere allerede eksisterende løsninger knyttet til emnet, og der bliver redegjort for teori, teknologi og empiri der kan være relevant for vores løsning.

3.1 Allerede-eksisterende løsninger

Som tidligere nævnt i rapporten er den økonomiske faktor særligt relevant for vores målgruppe, og det er derfor nærliggende at undersøge hvordan lignende løsninger fungerer. Da mange af disse løsninger findes på apps til smartphones, og de derfor også er nyttige og let tilgængelige, er det dér vi vil lægge vores fokus i dette afsnit.

Udvalget af apps, der tilbyder værktøjer til at budgettere mad og lave madplaner, spænder meget bredt. Ved en simpel søgning på App Store med nøgleordet ”madplan” dukker der en masse forskellige madplan-apps op. De første tre madplan-apps der dukker op er heriblandt ”Mad For Fattigrøve af Fie”, ”Hofretter” og ”Valdemarsro”. Ved første øjekast over app-beskrivelserne, kan det hurtigt konstateres at de har et fælles formål om at gøre det nemmere at planlægge måltiderne i dagligdagen. I dette projekt har vi valgt at fokusere på at analysere tre apps: ”Mad for fattigrøve af Fie”, ”Sulten” og ”Too good to go”. Disse apps har alle, i varierende grad, fokus på at spare penge samtidig med at man reducerer madspild.

3.2 Mad for fattigrøve af Fie

Appen ”Mad For Fattigrøve af Fie” er udviklet af Fie Fabricius Winther, som selv var i målgruppen 20-34 år (i en alder af 25 år), da app’en blev udviklet. Fie Winther ønsker ”at gøre en forskel - og med de erfaringer jeg selv har gjort mig på en lav indkomst, så er jeg nødt til at give min viden videre”. Formålet med ”Mad For Fattigrøve af Fie” er at hjælpe folk, der

har en forholdsvis lav indkomst svarende til SU-niveau, med at skabe en sammenhængende madplan. Udgangspunktet i disse madplaner er fokus på billige råvarer, og i den forbindelse råvarer på tilbud. Det første man ser ved at åbne Fies app, er en liste over forskellige forslag til madplaner til én enkelt uge. De unikke funktioner kan tilgås i menuen, nemlig "Opskrifter", "Min madplan", "Tøm køleskabet" og "Allergener". Funktionerne gør det muligt for brugeren at hente inspiration til forskellige retter, samt at tage forbehold for sine allergier over for bestemte ingredienser.

Når brugeren vælger at lave en madplan for hele ugen, laver appen en indkøbsliste baseret på hvad de har og hvad de mangler. Funktionen "Tøm køleskabet" tager udgangspunkt i de råvarer, man allerede har i køleskabet, og er tilsyneladende en funktion der er skabt til at motivere brugeren til at bruge hvad de har på hånden her og nu fremfor at handle unødigt ind. Det skal dog nævnes, at lige netop denne funktion er låst bag en paywall, og vi er derfor usikre på hvordan den fungerer i praksis. Ikke desto mindre er appen godt udstyret med overskuelige funktioner og informationer, der hjælper med at skabe et overblik over indkøb og indhold i køkkenet.

Kort opsummeret er hovedfokuset for "Mad For Fattigrøve af Fie" at spare på økonomien, og altså ikke som sådan hverken madspild eller miljøvenlighed. Man kan dog argumentere for, at selve det at tilrettelægge og planlægge madlavning på en struktureret og økonomisk forsvarlig måde vil bidrage til et mindre madspild. Som tidligere nævnt i rapporten er det anbefalet at man netop planlægger sine måltider så detaljeret som muligt, da dette i store træk skaber mindre overskydende mad.

3.3 Sulten

"Sulten" er en norsk app som endnu ikke er kommet på den danske app store, men som i øjeblikket er ved at blive oversat til dansk til brug på det danske marked. Appen er lavet af Karen Elene Thorsen, som i Norge er kendt under navnet "fattig.student" på Instagram. På sin Instagram-side delte hun opskrifter og tips specifikt til fattige studerende, og blev rasende populær på dette. Det førte til udviklingen af app'en, og på hjemmesiden sulten.app. Karen Thorsen mener at "Sulten skal hjælpe dig med at spare penge, smide mindre mad ud og inspirere dig til at tage bæredygtige valg for kloden" **sulten** Ved første øjekast er appen altså stærkt relevant for vores problemformuler-

ing, da den berører både besparelser, madspild og bæredygtighed, og det afspejler sig også i de funktioner app'en har.

App'en består af en søgefunktion hvor man kan søge på opskrifter og madplaner, hvilket minder meget om app'en fra tidligere kalder "Mad For Fattigrøve af Fie". Inde i app'ens funktioner finder man også et arkiv med madplaner der kan tilrettelægges efter behov, som f.eks. vegetarmad, pescetar, mv. Herudover findes der tips til opskrifter, der gør brug af typisk restemad.

Stryger man til næste side kan man lægge varer til i en indkøbsliste, og hvis man stryger videre endnu finder man funktionen "Udforsk". "Udforsk" spørger "Hvad du har i køleskabet?", og her kan der søges på forskellige ingredienser, man har tilgængelig, og danne en liste over opskrifter som bruger disse ingredienser. Disse opskrifter er dog låst bag en paywall, så dem har vi ikke kunnet undersøge.

Ligesom med "Mad For Fattigrøve af Fie" er "Sulten" altså direkte henvendt til folk, der ønsker at spare på både økonomien og madspildet. En mærkbar forskel er dog, at "Sulten" tilbyder opskrifter, der benytter de rester man har tilgængelig.

3.4 Too Good To Go

"Too Good To Go", benævnt som TGTG, er en app der skiller sig ud fra de andre apps. App'en tilbyder ikke opskrifter eller madplaner, men har fokus på at reducere madspild ved at sælge overskudsmad fra detailhandlen og restaurationsbranchen, såsom dagligvarebutikker, cafeer, hoteller osv. TGTG har et direkte samarbejde med flere aktører i fødevareforsyningskæden, og bidrager således til at mindske madspild på flere niveauer udover i den private husholdning. TGTG er altså et eksempel på en løsning, der formår at bringe flere aktører i spil på en måde der gavner alle involveret på en bæredygtig måde - detailhandlen og restaurationsbranchen formindsker sit madspild, og brugeren sparer på økonomien.

Ulempen ved TGTG er imidlertid, at man som bruger ikke selv kan vælge, hvad det er man køber. Når man køber en pose fødevarer fra f.eks. Netto eller lignende via appen, er indholdet ikke beskrevet i detaljer, men blot med en kategori. Det kunne eksempelvis være "Frugt og grønt" eller "Brød". Når

man derfor køber en pose fødevarer ved man ikke, om indholdet er noget man vil spise, eller om man mister interessen. Meget af indholdet er da også datovarer, der allerede er ved at overskride sidste brugsdato, og det er ikke utænkeligt at brugeren vælger ikke at spise dem hvis varerne tilmed ikke engang falder i deres smag. Et andet problem med at indholdet er datovarer er, at mængden hurtigt kan være for meget (eller for lidt), og at brugeren måske ender med at smide fødevarerne ud før de når at bruge dem. Det kan dog omvendt være en fordel, at brugeren netop ikke skal tage stilling til andet, end hvilken type pose med fødevarer, de vil købe. Hvis man allerede har en travl hverdag er der tid og penge at spare, hvis man bare skal ind og hente en færdigpakket pose fødevarer i f.eks. SuperBrugsen.

TGTG er således ikke skræddersyet til den enkelte bruger, som "Mad For Fattigrøve af Fie" og "Sulten" kan være. Det er heller ikke muligt at tilrettelægge en madplan, og det ville ikke være nyttigt for brugeren at have en sådan funktion når indkøbet er ukendt. Omvendt kan det også være en fordel for nogle brugere, især hvis man ikke har behov for en specifik kost og ikke har noget imod at købe billigt og tilfældigt. En anden forskel er naturligvis også, at man køber fødevarer direkte igennem appen, fremfor at lave en madplan eller lignende hvor man beskriver hvad man skal købe.

3.5 Ligheder og forskelle mellem eksisterende løsninger

De mest mærkbare forskelle findes især mellem appen "TGTG", og de to madplans-apps "Sulten" og "Mad For Fattigrøve af Fie". De to madplans-apps deler mange ligheder, både i funktionalitet og hensigt, der primært handler om at tilrettelægge kost og indkøb på en billig og bæredygtig måde. De er udstyret med en del hjælpemidler til den enkelte bruger, der er til for at skabe et overblik som på sigt kan lede brugeren hen på nogle mere økonomi-venlige og bæredygtige kostvaner. Her er det til gengæld brugerens eget ansvar både at skaffe de pågældende varer, og udøve det modicum af selvdisciplin og overskud der skal til for at holde en madplan. Disse apps kræver, for at de skal bruges optimalt, et forholdsvis højt bruger-engagement for at hensigten med apps'ne skal opfyldes. At holde en madplan for blot en uge kræver allerede et vist overblik, som det ikke er alle potentielle brugere der har. Især målgruppen, der er fokus for denne rapport, har en uforudsigelig hverdag der resulterer i spisevaner der ikke følger et fast mønster. Det kan dog sagtens argumenteres, at bare det at have en madplan man kan følge,

vil bidrage med at skabe sunde spisevaner, og om ikke andet kan madplanen tjene til inspiration for den enkelte.

Hvor de to madplans-apps kræver et højt bruger-engagement er "TGTG" til gengæld forholdsvis simpelt at bruge - og kræver langt mindre engagement. Hensigten med appen er jo ganske enkelt at sælge ud af mad og datovarer fra detailhandlen, og det er op til brugeren hvorfra og hvor ofte de vil gøre et køb. Selvom fokus ikke er brugerens egen kost, er appen alligevel med til at mindske madspild på en økonomisk- og miljømæssigt bæredygtig måde. Man kan nemt skaffe sig en pose indkøb og mad, der ellers ville have kostet det dobbelte af prisen hvis ikke man købte igennem TGTG. TGTG er desuden den eneste app af de tre, der af sig selv har en direkte positiv effekt på miljøet og madspild ved at sælge ud kun af datovarer, der ellers ville være gået til spilde.

4 Analyse

Dette kapitel vil handle mere specifikt om den information og analyse, der skal til for at vi kan undersøge og understøtte en mulig løsning til problemformuleringen. Vi vil starte med at overveje og sammenligne forskellige løsninger, med fokus på allerede eksisterende produkter. Derudover, præsenteres vores spørgeskema, målgruppeanalyse og persona, som fører os til en analyse af en række scenarier. Målet med dette kapitel er hovedsageligt at identificere forskellige krav til en mulig programløsning, samt at indskrænke vores problemformulering yderligere.

4.1 Spørgeundersøgelse

Som tidligere nævnt i rapporten er undersøgelsen fra DCA, der undersøgte danskernes madspild i detaljer, meget relevant for vores problemstilling. Rapporten adspurgte forskellige aldersgrupper om deres madspild, og undersøgte de underliggende tendenser i folks mad- og spisevaner. Den havde til gengæld ikke så stor fokus på den målgruppe, som denne rapport arbejder med - navnlig de unge voksne i alderen 18-34. Hovedsageligt var det ældre voksne, der deltog i undersøgelsen fra DCA, og derfor valgte vi at lave vores eget spørgeskema. I dette afsnit vil vi præsentere resultatet af vores undersøgelse og diskutere, hvordan dette kan have relevans for vores problemformulering.

Spørgeskemaet blev uddelt til vores medstuderende her på 1. semester af software-linjen på Aalborg Universitet i København, og vi fik i alt 28 svar. Alle de adspurgte hørte til i den ønskede målgruppe.

Undersøgelsen viste også at 32.1% svarede at de ikke aktivt forsøger at reducere deres madspild. Når en tredjedel af vores målgruppe ikke aktivt vælger at reducere deres madspild, bekræfter det at der et relevant problem der skal tages hånd om.

Vi spurgte så om hvor ofte de smider store mængder mad ud. Her svarede 14.3% af vores deltagere at de tit smider store mængder mad ud. Derudover smider 67.9% af deltagerne sjældent store mængder mad ud.

Når deltagerne både svarer, at hele 32.1% ikke aktivt forsøger at mindske sit madspild, og at det kun er 17.9%, der aldrig smider store mængder mad ud, må det med en vis sikkerhed kunne antages at de fleste i denne

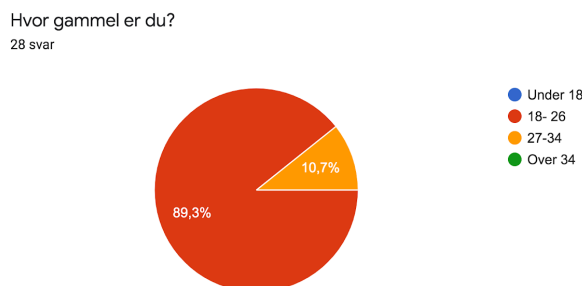


Figure 4: Indsamling af information om alder, fra eget spørgeskema

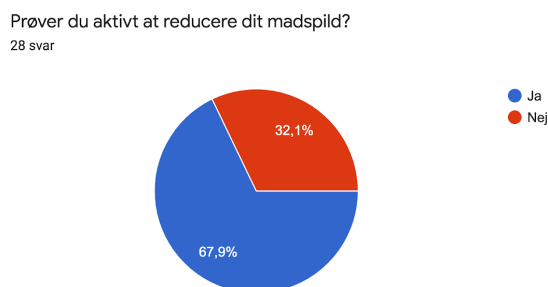


Figure 5: Indsamling af information af aktivitet i forhold til madspild, fra eget spørgeskema

målgruppe jævnligt smider mad ud. Der vil naturligvis altid være en form for påvirkning af egen selvopfattelse i disse spørgeskemaer, der gør at tallene ikke nødvendigvis repræsenterer virkeligheden så nøjagtigt endda. Det er ikke utænkeligt at der bliver smidt en god del mere mad ud end tallene udtrykker, og som tidligere nævnt i rapporten har de fleste mennesker en tendens til at undervurdere sit eget madspild - ikke mindst fordi mange har en misforstået opfattelse af hvad madspild er.

Omvendt viser det faktum, at både 67.9% af de adspurgte aktivt forsøger at reducere deres madspild, og at 67.9% også sjældent smider store mængder mad, at en stor del af målgruppen dog alligevel er opmærksomme og i nogen grad forsøger at forhindre at fødevarer går til spilde. Ikke desto mindre er det

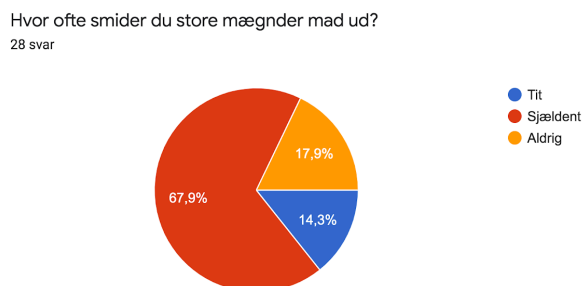


Figure 6: Indsamling af informaiton of madspild, fra eget spørgeskema

tydeligt at se, i tandem med de tidligere nævnte undersøgelser, at madspild er et problem for de unge voksne.

I samme spørgeskema spurgte vi deltagerne om hvilke incitamenter, der ville motivere dem til at reducere deres eget madspild. Her kunne man vælge flere muligheder, og et overvejende flertal svarede at spare penge ville være en motivator med 85.7%. Næst efter dette valgte 67.9% at hjælpe miljøet, og herefter valgte 57.1% tidsbesparelse som en motivator.

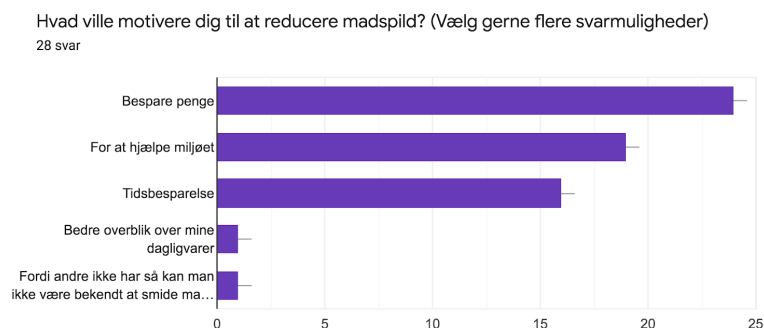


Figure 7: Indsamling af information of motivationer i forhold til madspild, fra eget spørgeskema

Når det kommer til indkøb svarede 50% at de kun nogen gange køber madvarer ind på impuls, 25% svarede de købte ofte ind på impuls, og de resterende

25% svarede at de sjældent eller aldrig køber fødevarer på impuls. Derudover spurgte vi hvor ofte de endte op med ekstra madvarer, de ikke ved hvordan de skal bruge, og her var resultatet at 46.6% nogen gange endte med ekstra madvarer. Til sammenligning ender 28.6% ofte med ekstra madvarer, og de resterende 25% ender sjældent eller aldrig med ekstra madvarer de ikke ved hvordan de skal bruge.

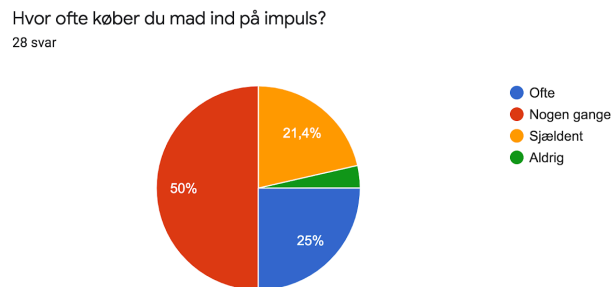


Figure 8: Indsamling af informaiton of madspild, fra eget spørgeskema



Figure 9: Indsamling af informaiton of madspild, fra eget spørgeskema

Dette, sammen med de andre undersøgelser vi har præsenteret i denne rapport, bekræfter at unge har en tendens til at købe ting på impuls, og at de ofte eller nogle gange ender op med ekstra madvarer de ikke ved hvordan de skal bruge. Det kan tænkes at dette er én af hovedårsagerne til meget af det madspild, der sker blandt unge voksne i dag. Når det kommer til madplan svarede kun 3.6% af deltagerne, at de altid følger en madplan, mens hele 53%

svarede at de aldrig følger en madplan. Resten af deltagerne svarede, at de kun følger en madplan nogle gange.

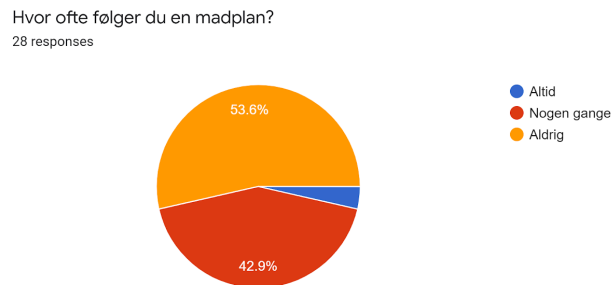


Figure 10: Indsamling af informaiton of madspild, fra eget spørgeskema

Til sidst spurgte vi deltagerne om hvad de føler, de har mest brug for mellem to valgmuligheder. Enten 1) en fast ugentlig madplan, eller 2) opskrifter der er tilpasset de ingredienser (og rester) man har til rådighed. Her svarede hele 75%, at de har mest brug for tilpassede opskrifter, hvor 25% svarede at de har mest brug for digital madplan.

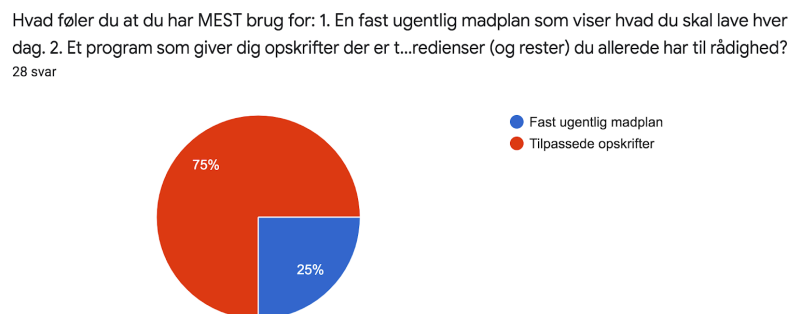


Figure 11: Indsamling af informaiton of madspild, fra eget spørgeskema

Disse resultater kunne tyde på, at målgruppen ville have mest gavn af en løsning der er skræddersyet til den enkeltes behov således at det kun er få ting, brugeren skal tage stilling til. Man kan ræsonnere sig frem til, set i lyset af de unge voksnes travle hverdag, at en fast madplan kræver for meget tid og energi at skulle vedligeholde for eksempelvis studerende i målgruppen.

4.2 Konklusion af Spørgeundersøgelse

Der kan konkluderes tre hovedpunkter fra spørgeundersøgelsen med målgruppen unge voksne alderen 18 - 34 år:

De smider jævnligt mad ud, selvom langt de fleste aktivt forsøger at reducere mængden af madspild. De køber jævnligt mad ind via impulskøb, og sidder ofte inde med rester i køleskabet. De færreste følger en madplan, og når de gør er det ofte kun sporadisk. Skræddersyet opskrifter alt efter hvad de sidder inde med ville være at foretrække.

De to første hovedpunkter illustrerer altså problemstillinger, som andre rapporter og undersøgelser har bekræftet før. Det tredje hovedpunkt tyder på, at målgruppen på trods af at de ikke så ofte følger en madplan meget gerne vil have værktøjer til at lave mad - fortrinsvis af ingredienser, de allerede har. Det må antages, at i en travl hverdag er der brug for en større fleksibilitet, der kan hjælpe med at danne et løbende overblik.

Funktioner og værktøjer der tager udgangspunkt i hvad man har i køleskabet, og danner opskrifter baseret på dette, er i de tidligere nævnte digitale løsninger "Mad For Fattigrøve af Fie" og "Sulten" låst bag en betalingservice. Det er ikke utænkeligt, at målgruppen ville have størst gavn af netop en sådan funktion, da den på sigt ville kunne opfylde tre behov for målgruppen:

At mindske madspild og bruge ingredienser (og rester), de allerede har At bruge færre penge på mad og impulskøb At kunne få opskrifter og vejledning fra dag til dag, eller sågar måltid til måltid, fremfor at skulle planlægge en hel uge.

At bygge en tilpasset løsning, der kun netop drejer sig om disse tre behov, kunne altså være en måde at angribe problemerne angående de unge voksnes store madspild. Hvis man skal indskrænke problemet yderligere, handler det i bund og grund om at gøre unge voksne i stand til først at bruge hvad de har i køleskabet eller køkkenet, og dernæst købe ind alt efter hvad de mangler. Sådan en tilgang passer godt til en sporadisk hverdag, der gør planlægning svær, og det giver økonomisk god mening hele tiden at kunne basere sine måltider på det man allerede har.

4.3 Målgruppeanalyse

Med denne målgruppeanalyse beskriver vi i detaljer de generelle kendetegn for de unge voksne i alderen 18 - 34 år. Da vi i udgangspunktet stiler efter at bygge et program, der har til formål at mindske madspild gennem et økonomisk incitament, skal vi først finde ud hvilke personer i den gruppe der har størst gavn af at spare. Vores formodning er studerende på SU, da de ofte sidder inde med den laveste indkomst. Dermed ikke sagt, at vores løsning ikke også ville kunne egne sig til personer udenfor målgruppen.

Kendetegn for folk i alderen 18 - 34

Studerende, løbende karriereskift, "crash-and-learn" periode der er præget af nogle meget forskelligartede år, hvor man danner sig en "voksen" identitet. Nogen stifter også familie i disse år. Henover årene kommer der desuden rent økonomisk til at ske et stort spring, hvor man går fra en forholdsvis lav indkomst til en langt højere indkomst.

Traditionelt set er den yngre generation mere fokuseret på bæredygtighed end de ældre, men dog er der i Danmark på et bredt plan enighed om at der skal sættes aktivt ind mod klimaforandringerne, hvor en del industrier bidrager til udfordringer herfor.

Dog er de unge voksne den befolkningsgruppe, der både er mest bekymrede for klimaforandringerne[17], samtidigt med at de er den målgruppe der smider allermest mad ud fra køkkenet[14].

Til gengæld benytter de unge voksne i højere grad apps for at forhindre madspild, som f.eks TooGoodToGo, YourLocal, Mad for fattigrøve af Fie o.l.[18] Det er derfor ikke irrelevant at forsøge at lave et program, der kan sætte ind og hjælpe med at strukturere målgruppens indkøbs- og madvaner.

Økonomi

	20 – 24	25 – 29	30 – 34
Mænd	10.006,69 kr.	15.656,54 kr.	19.792,32 kr.
Kvinder	8.968,34 kr.	13.814,69 kr.	16.719,71 kr.

Figure 12: Udbetalt A-indkomst pr. måned, i gennemsnit for de 22 studiebyer. Data er taget fra Danmarks Statistik.[19]

Hvis man tager udgangspunkt i f.eks. gruppen 20 – 24 årige, hvor der findes flest studerende, ligger den tilgængelige indkomst altså mellem 8.000 – 10.000 i gennemsnit. I de fire største studiebyer København, Aarhus, Aalborg og Odense ligger de studerendes husleje på omkring 3.000 – 5.000 for et værelse.[20] En person mellem 20 – 24 år, der bor i én af de store studiebyer, og som bor til leje på et værelse, har derfor i gennemsnit et budget rangerende fra 3.000 – 7.000 til alt udover husleje. Vores formodning om at SU-studerende sidder inde med en lav indkomst holder altså stik. At fokusere på denne gruppe – eller i hvert fald at indrette vores løsning til dem – giver god mening idet denne gruppe økonomisk er den mest sårbare, og derfor kunne have størst gavn af en program-løsning der fokuserer på at mindske madspild gennem et økonomisk incitament. Som tidligere nævnt i rapporten ville den største motivation for at mindske madspild for målgruppen også være at spare penge, idet madspild kan sidestilles med at spille penge.

Indkøbsvaner

[14] Man kan som udgangspunkt tale om, at der er to særlige indkøbsvaner der ligger til grund for meget af madspild for 20 – 34 årige:

Impulskøb

Unge indkøbsvaner er præget af spontane impulskøb, og det er der en række årsager til. De har en travl hverdag, der ofte er uden fast struktur. Planlægning af hvornår et måltid skal ligge er derfor ofte umulig, og de har ikke et fast spisemønster.

Med den ustrukturerede hverdag følger et mistet overblik over hvor, hvornår

og hvad, man spiser, og de nemmeste løsninger bliver derfor take away, kantine mad, eller hurtigt indkøb på farten.

Mængdeindkøb

Fordi de unge voksne netop har en uforudsigelig hverdag og dog stadig skal købe ind, går de gerne efter tilbudsrabatter og mængdeindkøb. Den ustrukturerede hverdag skaber imidlertid stadig problemer, da man ofte ikke når at få brugt de indkøb man har lavet, og der bliver derfor begået en del madspild. Den manglende balance i planlægning og indkøb fører således tilbage til impuls køb igen, der virker som den lette lappeløsning.

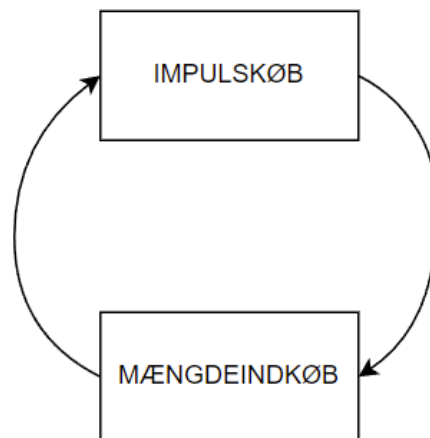


Figure 13: Impuls og mængdeindkøb diagram

Der er tydeligt et behov for remedier til at bryde "den onde cirkel", der illustreres af unge voksnes indkøbsvaner foroven. Som nævnt tidligere vil den løsning, dette projekt fokusere på, være at hjælpe unge voksne med så vidt muligt at benytte sig af de ingredienser de allerede sidder inde med.

4.4 Den nye problemformulering

Ovenstående spørgeundersøgelse og målgruppeanalyse leder os hen til et indskrænket problemfelt, der yderligere præcisere vores problemformulering. Problemformuleringen fremad i teksten lyder altså:

”Unge voksne alderen 20-34 er den målgruppe, der begår mest madspild, og det sker for det meste ved at der bliver smidt meget mad ud fra køleskabet. Dette skyldes ofte et mistet overblik over hvad de har af indkøb, og en uforudsigelig hverdag.

Hvordan kan man vha. et program, der fokusere på at brugeren skal spise indholdet i deres køleskab, hjælpe unge voksne med at få et bedre overblik og tømme køleskabet fremfor at smide mad ud?

4.5 Det potentielle program

Hvis man som udgangspunkt skal bygge et program, der bryder den onde indkøbs-cirkel og opfylder den enkelte brugers behov for tilrettelagte opskrifter, kan man snakke om at der er brug for i hvert fald tre potentielle funktioner:

En funktion, der kan vise brugeren hvad de har stående af ingredienser i køleskabet og/eller køkkenet. En anden funktion, der viser brugeren de opskrifter vedkommende kan lave med de ting, de har liggende. En tredje funktion, der kan vise hvad brugeren kun lige akkurat behøver at købe ind for at kunne lave en opskrift baseret på hvad de i forvejen har.

Sammenlagt ville disse funktioner udgøre et program, der måske ville kunne løse den meget specifikke problemstilling om at gøre unge voksne bedre i stand til at gøre brug af sine indkøb derhjemme. Det er ikke utænkeligt at man kunne koble flere funktioner til et sådant program. Da vi imidlertid har behov for at undersøge, om der kunne være potentiale i blot disse tre funktioner, vil vi basere en scenarieanalyse på de tre ovenstående funktioner.

I den følgende scenarieanalyse vil disse tre funktioner indgå i en kontekst, hvor det forestilles hvordan de ville fungere som et færdigt program, der ville kunne bruges ud i virkeligheden. Navnene på de tre funktioner vil være som følger:

“Digitalt køleskab” - viser brugeren hvad de har stående af ingredienser i køleskabet og/eller køkkenet. “Opskriftsliste” - viser brugeren de opskrifter vedkommende kan lave med de ting, de har liggende. “Indkøbsliste” - viser hvad brugeren kun lige akkurat behøver at købe ind for at kunne lave en opskrift baseret på hvad de i forvejen har.

4.6 Persona

Nedenstående persona er en sammensat karakter af de 18 - 34 årige, og skal tolkes som et tænkt eksempel på hvordan en person fra denne målgruppe kunne være. Formålet med personaet er således både at illustrere en person fra målgruppen, og detaljere en fiktiv person der kan bruges i scenarier, hvor man undersøger funktionaliteten af programmet. Jo bedre man kender sin målgruppe, des mere udtænkte scenarier - og senere endnu, tests af prototyper - kan man lave som en del af projektet.[21]

Persona: Kenneth

Alder: 22 år

Beskæftigelse: Universitetsstuderende i Hovedstaden

Anden beskæftigelse: Studiejob/caféjob på ca. 8-12 timer om ugen

Budget til mad om måneden: ca. 1000-3000 DKK pr. måned, men det svinger ofte hvorvidt budgettet bliver holdt eller ej. Visse måneder er der desuden ikke lige så mange penge på kontoen, f.eks. pga. julegaver, opsparing til ferie, en ekstraregning, for mange sociale arrangementer osv.

Hvad køber Kenneth?

Kenneth laver ofte et meget stringent budget til mad, som han stiler efter at holde hver måned. Han køber derfor så billigt han kan, når han handler ind. Ofte skrider budgettet dog en smule, fordi Kenneth ikke altid har tid til at lave mad og bruge de ting han har i køleskabet. Dette skyldes bl.a. at Kenneth måske ikke når at tjekke hvad han har i køleskabet og derfor ikke har en klar idé om hvad han kan nå at lave af mad.

Overskuddet til at lave maden selv er der heller ikke altid pga. universitetssarbejde og lignende, og en generelt spontan hverdag, så det bliver til en del købte måltider i skolekantinen hvor prisen er sat forholdsvis høj sammenlignet med en hjemmelavet madpakke. Til og fra arbejde bliver der af og til også købt et enkelt måltid i f.eks. 7-11. Dette budgetterer han også for i sit madbudget, men det kan være svært at holde. Når Kenneth til gengæld har tid til at købe ind og planlægge, køber han gerne langtidsholdige varer, han kan smide i fryseren, såsom grøntsager, datomærket kød, eller rugbrød.

Her sigter han efter at købe mad til gul pris og datovarer, han enten kan fryse ned, eller som han er ret sikker på at han kan nå at spise inden de bliver dårlige. Heri lægger fælden til gengæld i de dage, hvor han ikke når at bruge dem, så f.eks. den billige karton mælk til 2 DKK er nået at blive dårlig. Kenneth forsøger så vidt han kan at have en stor beholdning af pasta, makrel på dåse, havregryn og andet, der lynhurtigt kan blive til et måltid mad. Når Kenneth så alligevel er ude og handle, kan han godt lide at købe en billig snack til sig selv som f.eks. en billig plade chokolade eller lignende. Kenneth forsøger at kombinere en klimavenlig måde at købe billigt ind på ved bl.a. kun at købe kød og kylling på gul pris, som han så fryser ned til senere brug i nogle store gryderetter. Det tager til gengæld længere tid at planlægge madlavning med de frosne varer, for så skal han have et større overblik over hvornår han skal bruge dem. Kenneth køber fortrinsvis ikke økologisk, fordi det som oftest er dyrere.

Hvad kunne Kenneth spise, hvis det skulle være helt billigt?

Hvis Kenneth KUN skulle spise ting, der både passede ind i budgettet og som han selv fik lavet aller-billigst, så ville han lave en stor beholdning af f.eks. linsegryderetter som han kunne fryse ned, og så have havregryn, rugbrød og pålæg liggende til morgenmad og frokost. Dette kunne skabe et større overskud af penge til måske at fyre af på et måltid ude i byen med vennerne, som han kunne købe med god samvittighed. Dette kræver til gengæld enormt meget mere planlægning og overblik, og da Kenneths planer ofter skifter fra dag til dag med både universitetsplanlægning og sociale arrangementer er det stort set umuligt. Vi benytter Kenneth i nogle scenarier senere, hvor han interagerer med vores program i en tænkt kontekst.

4.7 Scenarier, en fiktiv test med persona

Formålet med disse scenarier er at give indblik i hvordan en potentiel bruger i målgruppen ville interagere med vores program. Scenarierne er skrevet i en lineær tidslinje ved brug af tænkte eksempler, der involverer et persona som er repræsentativt for den gennemsnitlige bruger i vores målgruppe.

De forskellige præmisser udgør rammen for brug af vores potentielle program. I de to nedenstående scenarier bliver der således illustreret en lineær vej fra brugerens behov til programmets løsning understreget af events (hændelser), der viser hvordan interaktion mellem bruger og program løser behovet.

Undervejs i scenarierne er der indskudt fejl-scenarier, der pointerer steder i scenariernes forløb hvor brugeren kunne opleve problemer med programmet. Disse fejlscenarier er indrykket i teksten. Fejl-scenarierne er således potentielle breaks, hvor brugeren stopper med at bruge programmet pga. fejl i designet. Hvis ikke der sker sådan et break, fortsætter scenariet frem til succeskriteriet.

Succeskriteriet for begge scenarier er, at brugeren får lavet mad af de ting, de allerede har ved hånden i køleskabet - og ellers får købt minimalt ind til en bestemt opskrift i programmet.

Markeringen **EVENT** markerer en direkte interaktion mellem bruger og program, eller en handling som programmet foretager. Nedenfor er disse events illustreret vha. pile, både internt i programmet og som interaktion mellem program og bruger.

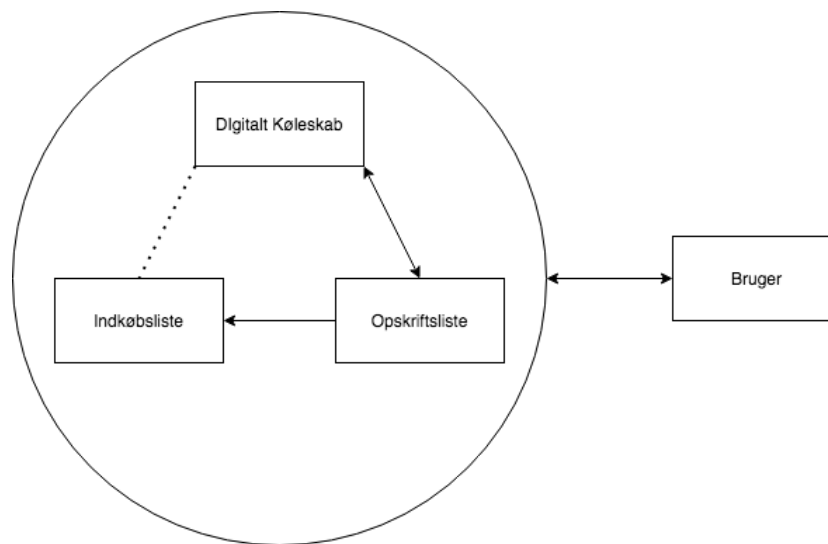


Figure 14: Diagram

Program-udgaven i disse scenarier har således de tre før-omtalte funktioner, som udgør vores potentielle program:

“Opskriftsliste” - viser brugeren de opskrifter vedkommende kan lave med de ting, de har liggende.

“Indkøbsliste” - viser hvad brugeren kun lige akkurat behøver at købe ind for at kunne lave en opskrift baseret på hvad de i forvejen har.

Når der i teksten i scenarierne bliver omtalt en funktion i programmet vil denne være omsluttet af citations-tegn - f.eks “digitalt køleskab”.

Scenarie 1: Spaghetti/Normal brug

Formål: Formålet med dette scenarie er at få indblik i den normale operation og brug af programmet, under ideelle og normale forhold for en bruger inden for vores målgruppe.

Persona: Kenneth

Pre-conditions: Kenneth har allerede brugt appen (mindst én gang) før, og har derfor skrevet sine ingredienser ind i det “digitale køleskab” på sin telefon. Han har således kendskab til hvordan programmet fungerer.

Dette scenarie bliver halvvejs inde splittet op i to forskellige linjer, som repræsenterer to forskellige veje en normal interaktion med programmet kan gå. Begge veje starter på de samme punkt i scenarie, hvor de så herefter går ud i hver deres unikke situation.

Setting the scene: Kenneth har lige fået fri fra arbejde/universitetet, til slut på dagen.

Kenneth bliver sulten efter endt arbejde og overvejer, hvad han skal have at spise. Han vil helst ikke spise ude, fordi det koster for mange penge. Hans budget er ved at blive overskredet pga. de mange impulskøb hos (convenience chain).

EVENT: Kenneth kan ikke huske hvad han har liggende derhjemme og hvad han kan lave af mad, men så tjekker han (programmet).

EVENT: Kenneth trykker på “giv opskrift” for at se, hvad han kan lave. Han får vist en opskrift han kan lave baseret på det, han har liggende derhjemme. Han får opskriften på spaghetti og kødboller.

Linje A

EVENT: Programmet viser Kenneth at han har alle ingredienserne til spaghetti og kødboller liggende, og han tager hjemad.

EVENT: Kenneth kommer hjem og trykker på “opskrifter”.

EVENT: Programmet viser opskriften, som også indeholder vejledningen til hvordan det skal laves.

Linje A slut: **Fejlscenarie**

Kenneth har imidlertid glemt at han faktisk allerede har spist sit kød og har ikke tastet det ind i “digital køleskab”. Han mangler derfor nogle af de ingredienser som programmet tror, at han har. Han gider ikke købe ind igen og bestiller en pizza.

Linje A: Fortsat

Kenneth laver mad.

EVENT: Programmet spørger Kenneth om han har lavet opskriften, og han trykker “ja”.

EVENT: Programmet trækker mængden af brugte ingredienser fra “digitalt køleskab”.

Linje A slut: **Fejlscenarie**

Kenneth blev ikke mæt af kun 300 gr spaghetti og ville have ønsket, at programmet kunne portions-fordele eller ændre mængden af én ingrediens, i stedet for at have faste mængder i opskriften. Han bliver sur og dropper at bruge programmet.

Linje A: Fortsat Kenneth føler sig tilfreds og er glad for, at han har spist mad uden at bruge penge

Linje A slut: Succes

Linje B

EVENT: Programmet viser Kenneth at han IKKE har alle ingredienser til spaghetti og kødboller liggende. Det viser, at han mangler spaghetti (300 gr) på en besked der hedder “indkøbsliste”.

Linje B slut: **Fejlscenarie** Kenneth gider faktisk ikke ud og handle, så han går i 7/11 i stedet. Behovet for øjeblikkelig tilfredsstillelse vinder over ønsket om at spare penge.

Linje B fortsat: Kenneth er ganske vist træt, men vil stadig gerne hellere spare penge end at lave et impulskøb. Kenneth tager hen til et supermarked for at købe ind.

Linje B slut: **Fejlscenarie** Kenneth når ganske hen i supermarkedet, men han ser en ovnklar frysepizza på tilbud og beslutter, at det er nemmere at købe den og sætte den i ovnen end at lave mad.

Linje B fortsat: Han køber kun hvad programmet viser ham at han mangler, og ikke andre ingredienser, så han køber spaghetti. **EVENT:** Kenneth indtaster de ingredienser han lige har købt ind i det “digitale køleskab”. Fortsat fra start af linje A → Succes.

Scenarie 1 slut.

Scenarie 2: Førstegangsopsætning

Formål: Formålet med dette scenarie er at undersøge hvordan førstegangsopsætning og brug af programmet af et individ i vores målgruppe kunne gå.

Persona: Kenneth

Pre-conditions: Kenneth har IKKE brugt programmet før, men har hørt om det og vil gerne se, om det virker for ham. Han anerkender nemlig, at han ikke er så god til at bruge de ting han køber ind, og det vil han gerne ændre på.

(Da vi ikke kan nå at programmere en vejledning til hvordan brugeren skal bruge programmet må vi antage, at Kenneth intuitivt bruger programmet

efter hensigten.)

Setting the scene: Kenneth sidder derhjemme i ro og mag efter universitetet og har downloadet programmet til sin telefon.

EVENT: Kenneth kigger på startskærmen, roder lidt rundt i de forskellige funktioner for sjov. Han finder ud af, at han for at kunne danne et ”digitalt køleskab” selv skal indtaste de ingredienser han har.

Fejlscenarie: Kenneth gider ikke lave det arbejde, der er i at selv indtaste sine ingredienser. Han lukker programmet og får det ikke brugt.

Fejlscenarie: Kenneth opdager, at programmet ikke kan give ham opskrifter der er tilrettelagt efter hans ernæringsbehov (f.eks. nøddeallergi, veganske opskrifter m.v.) Han vælger ikke at bruge programmet.

Scenarie fortsat:

Kenneth danner overblik over hvilke ingredienser han har i sit køkken.

EVENT: Kenneth indskriver de ingredienser han har i det ”digitale køleskab”, ifølge navn (eks. ”tomater”) og mængde (eks. ”1 kg”).

Kenneth sætter sig med nogle afleveringsopgaver, indtil det bliver aften og han er sulten.

Fejlscenarie: Kenneth glemmer at han har sat programmet op, fordi at programmet ikke sender en påmindelse om at han ingredienser liggende til brug i køleskabet. Han bestiller en pizza i stedet.

Scenarie fortsat:

EVENT: Kenneth husker at han har sat programmet op med ingredienser, og vælger at gå ud i køkkenet og lave en af de opskrifter der ligger til rådighed.

EVENT: Programmet viser opskriften og vejledningen til hvordan det skal laves. Kenneth laver mad.

EVENT: Programmet spørger Kenneth om han har lavet opskriften, og han trykker "ja".

EVENT: Programmet trækker mængden af brugte ingredienser fra "digitalt køleskab".

Kenneth føler sig tilfreds og er glad for, at han har spist mad uden at bruge penge.

Scenarie 2 slut

Konklusioner:

Her er en række potentielle problemer, der kan vanskeliggøre brugen af det potentielle program. Disse er konkluderet ud fra de potentielle fejlscenarier, der blev beskrevet.

Upkeep

Programmet kræver aktiv interaktion mellem bruger og funktioner, for at programmet skal kunne fungere optimalt. Det vil sige, at hver gang at brugeren f.eks. bruger af en madvare, som er indtastet i det digitale køleskab, skal det rettes i programmet sådan at programmet ikke "lyver" for brugeren når denne spørger om han eller hun har de ingredienser de skal bruge.

Dette gælder naturligvis kun de ingredienser som rent faktisk bliver brugt i de indtastede opskrifter, men ikke desto mindre kan dette absolut være et irritationsmoment. På lige fod med andre programmer ønskes så lidt upkeep, eller sagt på dansk, aktiv indtastning af funktioner som muligt.

Desuden kan selve det faktum, at programmet kræver en aktiv indsats med indtastninger, absolut afskrække en del brugere.

Bruger-motivation Det er svært at forudsige lige akkurat hvor meget man kan motivere brugeren, når det gælder både brugerens eget incitament til at benytte programmet og den form for interaktion programmet kræver for at fungere optimalt. Med de forholdsvist begrænsede funktioner, programmet har, kan brugeren således hverken blive påmindet om hvad de sidder inde

med via en pop-up funktion, eller på anden vis overbevise brugeren om at benytte funktionerne i programmet med lignende beskeder. F.eks. kunne man forestille sig at en besked der lyder “Det koster i gennemsnit 100,- mindre at lave mad hjemme end at spise ude” motivere brugeren til at undlade at spise ude.

Det er ligeledes svært at svare på, hvordan man skal kunne stoppe brugeren fra f.eks. at give efter for den umiddelbare trang til hurtig mad, eller andre lettere løsninger.

Mangelfuld opskriftsliste - ingen hensyn til særlige behov

At programmere adskillige opskriftslistes, der tager hensyn til brugerens særlige behov, ville desværre på nuværende tidspunkt kræve et større stykke arbejde end der er tid og evner til. Portionsmængderne kan ydermere føles misvisende, hvis det er at brugeren benytter de skrevne mængder men ikke får opfyldt sine behov ifht. mængden af mad.

4.8 Kravspecifikationer

Inden implementeringen af et program begynder, er det vigtigt at have en forståelse for hvad det skal udvikles til at kunne. Det samme gælder for dette program. I den forbindelse er der blevet lavet en kravspecifikation, som detaljerer de forskellige krav der bliver stillet til programmets egenskaber. I dette kapitel beskriver vi altså potentielle funktioner, og både de førmtalte funktioner fra scenarier samt nogle nye funktioner vil blive beskrevet i detaljer. Senere vil disse blive prioriteret

“Opskriftsliste” - Fremvisning af relevante opskrifter

Vores opskriftsliste skal inkludere pre-installerede opskrifter, samt hvilke ingredienser retten til de individuelle opskrifter indeholder, og i hvor store mængder. Samtidig skal der også være instruktioner til hvordan opskriften tilberedes, som brugeren kan læse.

Informationen om mængden af ingredienser i en ret skal bruges når brugeren laver en ret, da denne mængde af ingredienser så skal trækkes fra det digitale køleskab. Brugeren skal kunne spørge om en opskrift, og skal så derefter

kunne få udleveret en liste over hvad de kan lave, ud fra hvad der er i køleskabet. Dette kræver en sammenligning af hvad der er i køleskabet, og de forskellige opskrifter. Hvis ingen opskrift kan laves, skal indkøbsliste-funktionen køres (se længere nede i teksten).

“Digitalt Køleskab” - Opbevaring af ingredienser

Det digitale køleskab skal gemme en liste af alle de ingredienser som er i de forskellige opskrifter, samt hvor mange eller hvor meget af de individuelle ingredienser brugeren har på lager. Der skal kunne tilføjes mængder af ingredienser til dette køleskab af brugeren, i takt med at de køber ingredienser ind og får dem på lager. Samtidigt skal ingredienser kunne fjernes automatisk, når en opskrift bliver lavet fra opskriftslisten. Til sidst skal brugeren også manuelt kunne fjerne mængder af ingredienser, når enten ingredienser bliver for gamle eller brugt på andre opskrifter.

“Indkøbsliste” - Fremvisning af manglende ingredienser

Denne funktion skal bruge information fra både opskriftslisten og det digitale køleskab. Denne funktion skal køres når brugeren gerne vil have givet en opskrift de kan lave, men ikke har nok ingredienser i køleskabet til at kunne lave nogen opskrift.

Funktionen skal så herefter “kigge” i køleskabet og opskriftslisten for at finde ud af, hvilken opskrift er tættest på at kunne blive lavet. Altså den opskrift hvor der mangler mindst ingredienser. Denne opskrift skal så gives til brugeren, sammen med en liste over hvad de mangler.

“Udløbsdato-tjek”

Denne funktion skal køres, når brugeren tilføjer en ingrediens til køleskabet. Når der bliver tilføjet en ingrediens, skal denne funktion kunne registrere tidspunktet den bliver tilføjet til køleskabet, samt hvor lang tid der er til ingrediensen når sin udløbsdato. Når udløbsdatoen er nået, skal programmet selv fjerne ingrediensen fra listen over ingredienser i det digitale køleskab, og melde brugeren om at dette er gjort. Funktionen skal ved hjælp af comput-

erens/telefonens eget ur holde styr på hvornår udløbsdatoen er nået. Samtidigt skal denne funktion samarbejde med opskriftslisten. Når en ingrediens er tæt på at løbe ud, skal opskriftslisten foreslå opskrifter der bruger denne ingrediens, så den bliver brugt før udløbsdatoen.

“Kalorie-indtag”

Denne funktion vil holde styr på hvilket mad, der bliver fjernet fra det digitale køleskab, altså den mad som det blev antaget at brugeren spiser. Funktionen vil så optælle de kalorier indeholdt i de fødevarer, der bliver spist over en uge, og til sidst sammenligne den mængde kalorier der bliver indtaget med gennemsnittet.

“Kvittering-scanner”

Denne funktion vil kunne scanne den kvittering man får efter at have købt sine madvarer ind. Ud fra denne kvittering skal funktionen kunne finde ud af hvilke madvarer der er blevet købt ind, og i hvilke mængder, og derefter indskrive dem i det digitale køleskab.

Formålet med denne funktion vil være at gøre det nemmere for brugeren at indsætte deres ingredienser i køleskabet, så de ikke ville skulle indskrive alt manuelt. Dette vil forbedre brugeroplevelsen af programmet, da brugeren vil spare tid.

4.9 Prioritering af krav (MosCow)

Ethvert projekt er begrænset af den mængde tid der er afsat til det, og det er en realitet at ikke alle ønskede funktioner og egenskaber kan blive udviklet i et projekt. Derfor skal de forskellige ønskede dele af programmet prioriteres, så det vigtigste bliver til en realitet. I dette projekt bliver der kravspecificeret via MoSCoW modellen, som er en model der inddeler krav til projektet i fire forskellige prioriteringsniveauer. “Must Have”, “Should Have”, “Could Have”, og “Won’t have”.

Must have

“Opskriftsliste” - Fremvisning af relevante opskrifter

“Digitalt Køleskab” - Opbevaring af ingredienser

Disse to krav er prioriteret som “Must Have”, da de er helt essentielle for programmets virkning. Uden disse to kan programmet ikke fuldføre sin mest basale funktion, som jo er at hjælpe brugeren med at holde styr på indholdet af sit køleskab og hjælpe vedkommende med at bruge af sine ingredienser. Derudover blev det også vurderet, at implementering af disse to i programmet ville være forholdsvis lette i forhold til andre krav, og skulle derfor fokuseres på først.

Should have

“Indkøbsliste” - Fremvisning af manglende ingredienser

Dette krav er prioriteret som “Should Have”, da dens implementering markant ville forbedre projektets evne til at besvare vores problemformulering. Dog er den stadigvæk ikke er absolut nødvendig, på samme måde som de to foregående funktioner. Hvis dette krav bliver opfyldt, bliver vores løsning dog forbedret.

Could have

“Udløbsdato-tjek”

Dette krav er prioriteret som “Could Have”, da det blev vurderet at fuldføre den ville være meget teknisk og tidsmæssigt krævende. Da den samtidigt ikke er essentiel til programmets funktion, og ville tage en masse ressourcer fra mere essentielle krav, har den fået denne lave prioritering.

Wont't have

“Kalorie-indtag”

“Kvittering-scanner”

Disse to krav er prioriteret som “Won't have”, altså som funktioner der ikke vil blive implementeret i programmet. Kalorie-indtag prioriteres som “Won't

have”, fordi at den ikke høre under programmets primære funktion som en funktion, der skal hjælpe med at reducere madspild, da denne funktion i stedet handler mere om sundhed.

Kvitterings-scanner er prioriteret som et “Won’t have” fordi at det blev vurderet af implementeringen af denne funktion ville tage for lang tid i forhold til hvad det bidrog til projektet, taget i betragtning at det sandsynligvis ville tage lang tid at implementere. Derudover ville det også kræve hardware, der ikke på nuværende tidspunkt er tilgængelig.

5 Design og implementering

I dette afsnit beskrives selve opbyggelsen af programmet, og der redegøres for valg af testmetoder og resultater. Funktionaliteten af programmet beskrives i detaljer med et flowchart og tekst med tilhørende billeder, der illustrerer hvordan koden til funktionerne tager sig ud i programmet. I den endelige udgave af programmet er funktionerne som følger:

“Opskriftsliste” - Fremvisning af relevante opskrifter “Digitalt Køleskab” - Opbevaring af ingredienser

“Indkøbslisten” er ikke blevet til en funktion for sig selv, men er til gengæld omskrevet til en information inde i den pågældende opskrift, der viser hvad brugeren mangler.

5.1 Flowchart over design

Da vi havde bekræftet hvilke funktioner der skulle benyttes i programmet, valgte vi at lave et flowchart. Formålet med dette er at få et visuelt overblik over funktionaliteten af vores program, samt strukturen af brugerfladen.

Ud fra dette flowchart kan man se den overordnet struktur af vores program, samt fremgangsmåden og samspillet mellem de forskellige funktioner af programmet. Alt fra menu-navigation til hvornår arrays skal inddrages kan ses via dette flowchart, og hjælp os med at fordele opgaverne. Før programmet giver nogen form for visuelt feedback til brugeren, læser det det nødvendige data først. Denne data, i form af et tekst dokument, skal indeholde vores ingrediensliste i vores digitale køleskab, og skal læses ind i et array som redigeres undervejs, og til sidst gemmes i et .txt dokument igen før programmet lukkes.

Vi ville gerne have at programmet skulle starte med en hovedmenu, som giver adgang til diverse funktioner i programmet uden at skulle følge en fast rækkefølge. I tilfælde af at brugeren ikke behøver at redigere det digital køleskabs indhold, har de muligheden for direkte at gå til opskriftslisten. Det digitale køleskab giver mulighed for redigering af antallet af en given ingrediens, og i tilfælde af at det ikke skal gøres alligevel, sendes brugeren tilbage til hovedmenuen. Opskriftslisten skal vise en liste af retter, hvor brugeren skal have mulighed for at vælge en ret, og antallet af portioner. Når

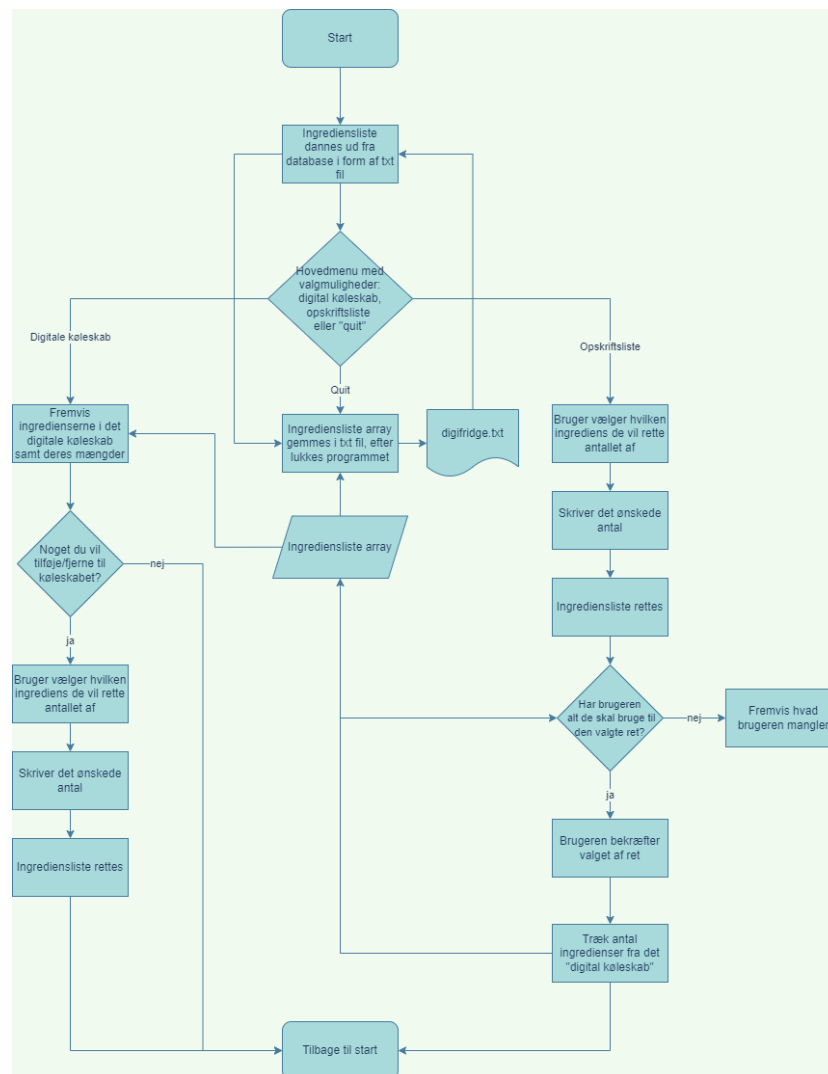


Figure 15: Flowchart over konceptet for det designede program

brugeren har bekræftet at de har valgt den ret, så skal antallet af ingredienser rettes til alt efter hvor mange ingredienser retten har krævet. I tilfælde af at brugerens ingrediensliste ikke opfylder kravene for at lave en givet ret, skal programmet kunne gøre brugeren opmærksom på hvad de mangler.

Den sidste valgmulighed er vores “quit” funktion, som bruger vores ingrediensliste-array og derefter gemmer det ved at udskrive det tilbage i dette samme

tekst-dokument som det læste fra i starten af programmet.

5.2 Implementering

Som det første i vores program deklarerer vi to arrays. Det første hedder “array” og indeholder mængden af ingredienser som datatypen double, hvilket gør det muligt for en ingrediens at have decimaler. Det andet array hedder “namearr” og indeholder navnene på ingredienserne. Derudover har dette array datatypen char, som gør det muligt for arrayet at indeholde karaktere.

```
//Main funktion
int main(void){
    //Array med antallet af given ingrediens
    double array[MAX_ING] = { 0 };
    //Array med navn for given ingrediens
    char namearr[MAX_ING][MAX_SIZE];

    //Funktion der læser ingrediensliste som et array
    reader(array);
}
```

Figure 16: Main funktion i vores program

Efter disse to deklARATIONER, bliver der kaldt funktionen “reader”, som skal indlæse mængden af ingredienser fra vores tekstdokument til “array”.

Først åbnes vores database i form af en tekst-fil, som hedder “digifridge.txt”. Vi benytter en for-løkke som tæller fra 0 og op til antallet af ingredienser. Inde i denne løkke tildeler vi et tal fra vores database til en plads i vores array. Når vi har talt fra 0 og op til antallet af ingredienser, har vi også skrevet alle vores mængder af ingredienser ind i hver deres plads i vores array, som vi nu kan benytte i vores program. Når vi er færdige med at bruge filen husker vi at lukke den. Det næste skridt i programmet er vores hovedmenu.

Det første der sker i hovedmenuen er, at vi kalder en funktion ved navn “clrscr”, som er en forkortelse for clear screen. Denne funktion sletter alt tekst på skærmen, hvilket gør menu navigation mere overskueligt for brugeren. Efterfølgende deklarerer vi en variabel ved navn “input”, som har datatypen char. Derefter udskriver vi mulighederne for brugeren på skærmen, og beder


```
int reader(double *array){  
  
    //Skift navnet paa txt filen  
    FILE *fp = fopen("digifridge.txt", "r");  
    int i;  
  
    //laeser per tal i txt filen og giver den vaerdi til hver index i array  
    for (i = 0; i < MAX_ING; i++) {  
        fscanf(fp, "%lf", &array[i]);  
    }  
  
    //Printer vaerdien er hvert index ud, bruges til test  
    /*  
    for(int i = 0; i < MAX_ING; i++){  
        printf("Plads nummer %d: ", i);  
        printf("%d\n", array[i]);  
    }  
    */  
  
    //Lukker filen  
    fclose(fp);  
  
    return *array;  
}
```

Figure 17: Reader funktion i vores program

brugeren om at indtaste deres valg. For at bestemme hvilken valgmulighed der skal benyttes bruger vi switch cases. Et switch case tager en variabel og sammenligner det med cases, og hvis det stemmer overens med et case, så køres det stykke kode der står inde i det case. Alt efter hvilken valgmulighed brugeren vælger, så kaldes den relevante funktion. Hvis brugeren for eksempel gerne ville se det digitale køleskab, og redigere sin ingrediensliste, så skal brugeren trykke "1", hvorefter menuen for det digitale køleskab kommer frem.

Strukturen for opsætningen af menuen er delvist det samme her. Hvis brugeren gerne vil rette mængden af en ingrediens skal de blot vælge hvilken ingrediens, og derefter skrive det antal de har af den valget ingrediens. Hvis brugeren kunne tænke sig at starte forfra og indstille alle ingrediensers værdi til 0, så kan de trykke "2" og derefter vil alle pladserne i vores array sættes til 0.

Dette er vores opskriftsliste-funktion, som benytter samme struktur som vores hovedmenu. Diverse valgmuligheder præsenteres for brugeren hvorefter programmet afventer et input. Efter bliver den relevante funktion kaldt frem

```
int menu(double *array, char *namearr) {
    //Ryd terminal
    clrscr();

    //Input variabel
    char input;

    //Print menu til forskellige dele af programmet
    printf("Title placeholder\n");
    printf("Vaelg en af foelgende valgmuligheder:\n");
    printf("[1] - Digitale koeleskab\n");
    printf("[2] - Opskrifter\n");
    printf("[3] - Indkoebsliste\n");
    printf("[q] - Luk programmet\n");

    //Modtag input
    scanf("%c", &input);

    //Case alt efter input
    switch(input){
        case '1':
            fridgemenu(array, namearr);
            break;
        case '2':
            recipemenu(array, namearr);
            break;
        case '3':
            shopmenu(array);
            break;
        case 'q':
            savedata(array);
            break;
        default:
            menu(array, namearr);
    }
}
```

Figure 18: Menu funktion i vores program

```
//Udskriver valgmuligheder til brugeren
printf("Vaelg en af foelgende valgmuligheder:\n");
printf("[1] - Rediger antal af en ingrediens\n");
printf("[2] - Reset\n");
printf("[q] - Gaa tilbage til hovedmenuen\n");

//Modtag input fra bruger
scanf("%c", &input);

//Switch case alt efter bruger input
switch(input){
    case '1':
        printf("Indtast nummeret paa den ingrediens du vil rette antallet af\n");
        scanf("%d", &IngNr);

        printf("Hvad er antallet du har af denne ingrediens?\n");
        scanf("%lf", &antal);

        //Antalet gives til det svarende i vores ingrediens-array
        array[IngNr - 1] = antal;
        //Menuen kaldes igen for at vise de nye mængder af ingredienser
        fridgemenu(array, namearr);
        break;
    case '2':
        //Alle ingredienser sættes til 0 i antal
        for(int i = 1; i < MAX_ING; i++){
            array[i - 1] = 0;
        }
        fridgemenu(array, namearr);
        break;
    case 'q':
        //Gå tilbage til hovedmenu
        menu(array, namearr);
        break;
    default:
        //Hvis der indtastes noget ugyldigt så kaldes den nuværende menu igen
        fridgemenu(array, namearr);
}
```

Figure 19: Fridgemenu funktion i vores program

ved hjælp af et switch case.

Dette er et eksempel på en af vores opskrifter. Her har vi unikke variabler som repræsenterer ingrediens kravet for opskriften, og de benyttes blandt andet hvis man ikke opfylder kravene for at kunne lave retten. I dette tilfælde vil brugeren blive påmindet om dette ved hjælp af vores “printf”, som skriver teksten “Du mangler:” efterfulgt af ingrediensens navn og antallet der mangler.

Dette er den anden halvdel af en rets kode. Her får brugeren præsenteret en række valgmuligheder, som giver dem mulighed for at kunne bekræfte at de har lavet en ret, eller skifte antallet af portioner. Hvis antallet af portioner

```
void recipemenu(double *array, char *namearr){
    clrscr();
    char input;
    getchar();
    printf("Opskriftsliste\n");
    printf("Vaelg en af foelgende retter:\n");
    printf("[1] - Spaghetti Bolognese\n");
    printf("[2] - Chili con carne\n");
    printf("[3] - Lasagne\n");
    printf("[4] - Spaghetti Carbonara\n");
    printf("[5] - Vegetar Lasagne\n");
    printf("[q] - Gaa tilbage til hovedmenu\n");

    scanf("%c", &input);

    switch(input){
        case '1':
            spaghettibolo(array, namearr);
            break;
        case '2':
            chili(array, namearr);
            break;
        case '3':
            lasagne(array, namearr);
            break;
        case '4':
            spaghetticarb(array, namearr);
            break;
        case '5':
            veglasagne(array, namearr);
            break;
        case 'q':
            menu(array, namearr);
            break;
        default:
            recipemenu(array, namearr);
    }
}
```

Figure 20: Kode for opskriftsmenu

```

void spaghettiCarb(double *array, char *namearr){
    clrscr();
    char input;
    double portioner = 1;
    getchar();
    //Ingredienskval som variabler
    double pasta = 90, bacon = 40, aeg = 1, ost = 25;

    //Lykke der fortsætter så længe brugeren ikke trykker "q"
    do{
        clrscr();
        printf("Opskrift: Spaghetti Carbonara\n");
        printf("En portion:\nPasta - %.2lf\nBacon - %.2lf\nAeg - %.2lf\nOst - %.2lf\n", pasta, bacon, aeg, ost);
        printf("Fremgangsmetode :\n");
        printf("Kog pasta som anviset paa pakken. Steg bacon i oliven olie.Bland aeg, ost og peber i en skaal.\n");
        printf("Naar pastaen er faerdig kogt, lag den i panden med bacon.\nHeld aegblandingen i panden og bland sammen i ca 1 minut.\n");

        //Hvis man ikke opfylder kravene af en given ingrediens, så får brugeren det at vide
        if ((array[23] - pasta) < 0){
            printf("Du mangler: %.2lf pasta\n", (array[23] - pasta));
        }
        if ((array[1] - bacon) < 0){
            printf("Du mangler: %.2lf bacon\n", (array[1] - bacon));
        }
        if ((array[21] - ost) < 0){
            printf("Du mangler: %.2lf ost\n", (array[21] - ost));
        }
        if ((array[32] - aeg) < 0){
            printf("Du mangler: %.2lf aeg\n", (array[32] - aeg));
        }
    }
}

```

Figure 21: Kode for en opskrift

skal ændres, så ganges de variabler i funktionen der repræsenterer ingrediens kravet med antal portioner. Derefter køres den første halvdel af programmet igen, da hele denne funktion kører i en løkke som fortsætter indtil brugeren trykker “q”. Grunden til dette valg er, at vi undgår at deklarere ingrediens variablerne igen, da det ville resultere i at deres værdi nulstilles tilbage til deres originale værdi som svarer til én portionstørrelse til én person . Når brugeren har bekræftet at de har lavet en ret, trækker programmet antallet af ingredienser brugt fra deres tilsvarende plads i vores array der indeholder antallet af hver ingrediens. Efterfølgende sendes brugeren tilbage til vores hovedmenu.

Dette er vores “save” funktion, som gemmer vores data tilbage i tekst dokumentet som vi læste fra da vi åbnede programmet. Funktionen starter med at åbne det samme tekst dokument, dog denne gang i “write” tilstand og ikke “read” tilstand, hvilket gør det muligt for os at skrive ned i dokumentet i stedet for at læse fra det som vi gjorde tidligere. Der tjekkes via et “if-statement” om vi får en gyldig respons fra vores file pointer som åbnede dokumentet, og hvis ikke så påmindes brugeren om at der skete en fejl ved åbningen af filen. Derefter er der en løkke der tæller fra 0 og op til antallet af ingredienser, og for hver gang der tælles op udskrives en ingrediens fra vores array og ned til tekst dokumentet. Når løkken brydes har programmet gemt antallet af hver ingrediens ned i et tekstdokument, som så kan læses igen næste gang programmet køres. Herefter lukkes filen og programmet

```
printf("Tryk paa et af foelgende valgmuligheder:\n");
printf("[1] - Bekraeft at du har lavet denne ret\n");
printf("[2] - Skift antallet af portioner\n");
printf("[q] - Gaa tilbage til hovedmenu\n");

scanf("%c", &input);
switch (input){
case '1':
    //Traek ingredienser fra array
    array[23] -= pasta;
    array[1] -= bacon;
    array[21] -= ost;
    array[32] -= aeg;
    clrscr();
    printf("Dit digitale koeleskab er nu rettet\n");
    getchar();
    printf("Tryk enter for at gaa tilbage til hovedmenuen\n");
    getchar();
    menu(array, namearr);
    break;
case '2':
    //Gang ingredienser der kraeves med input
    clrscr();
    pasta = 90, bacon = 40, aeg = 1, ost = 25;
    printf("Hvor mange portioner vil du gerne have?\n");
    scanf("%lf", &portioner);
    pasta *= portioner;
    bacon *= portioner;
    aeg *= portioner;
    ost *= portioner;
    break;
```

Figure 22: Mere kode for en opskrift

slukkes.

5.3 Testtyper

Som del af udviklingen af programmet skal det selvfølgelig testes. Et program skal primært testes for generel anvendelighed og tilfredsstillelse gennem flere forskellige typer af tests. På trods af begrænset adgang til forskellige

```
int savedata(double *array){
    char *filename = "digifridge.txt";

    // open the file for writing
    FILE *fp = fopen(filename, "w");
    if (fp == NULL)
    {
        printf("Error opening the file %s", filename);
        return -1;
    }

    int i;

    //laeser per tal i txt filen og giver den vaerdi til hver index i array
    for (i = 0; i < MAX_ING; i++)
    {
        fprintf(fp, "%f\n", array[i]);
    }
    // close the file
    fclose(fp);

    return 0;
}
```

Figure 23: Kode for at gemme data fra program.

mennesker at test programmet på, er et forsøg lavet på testpersoner inde for programmets målgruppe. Testen sammenlignes med vores scenarier, og analyseres for nyttig viden til forbedringer til næste iteration af programmet. Pga. tidspres blev programmet kun testet på to.

Det bør nævnes, at hensigten var at teste på to måder - henholdsvis “over the shoulder”-metoden og “langtidsbrug”- metoden. Selvom begge metoder er beskrevet i dette afsnit, er det kun “over the shoulder”-metoden vi nåede. I et potentielt fremtidigt arbejde med programmet og følgende iterationer ville der lægges fokus på at benytte begge testmetoder. De står nærmere beskrevet forinden, og er sat i kontekst med vores program.

Test 1: Over the shoulder

Testdesign:

I denne test gives testpersonen en gennemgående forklaring af formålet med programmet og dets opbygning. Testpersonen skal forstå hvordan programmet virker på et designmæssigt niveau. Dog skal denne introduktion ikke

indeholde interaktion med selve programmet, men derimod skal introduktionen være fuldkommen verbal.

Herefter får testpersonen lov til at interagere med programmet, og bliver bedt om at indtaste nogle ingredienser og om at få programmet til at fremvise en opskrift. Testpersonen må ikke få hjælp til brug af programmet under denne interaktion. Mens dette sker skal en udvikler holde øje med hvordan testpersonen bruger programmet. Specifikt skal de holde øje med hvad testpersonen tror hver menu-interaktion gør, og hvilke steder i programmet de bliver forvirrede eller sidder fast. Til sidst foretages et interview med testpersonen, hvor udvikleren spørger ind til umiddelbare tanker til designet af programmet, og hvad testpersonen selv synes var nemt at bruge programmet til og hvor de blev forvirrede.

Formål:

Formålet med denne test er at finde ud af hvordan en person i vores målgruppe naturlig interagerer med programmet, og hvad de forventer at forskellige interaktioner gør. Ideelt set ville dette finde sted uden den indledende introduktion og forklaring fra en udvikler, men da programmet ikke har nogen form for tutorial er det nødvendigt at give testpersonen en introduktion. Samtidigt er håbet også at finde forskellige kilder til forvirring eller frustrerende design, som kan ændres til at være nemmer for brugeren at interagere med.

Test 2: Langtidsbrug**Testdesign:**

I denne test undersøges programmets virkning over længere tid. Der sørges for, at testpersonen har en grundig forståelse for hvordan programmet virker, hvordan de skal interagere med det, og hvad det kan bruges til. Herefter bliver testpersonen bedt om at bruge programmet over en periode på flere dage, hvor de indtaster de ingredienser de køber ind, og bruger programmet til at vælge opskrifter de skal lave. Efter denne periode af brug bliver testpersonen interviewet, og der bliver spurgt ind til om programmet virkede efter hensigten. Der tages stilling til hvorvidt programmet gjorde det nemmere for testpersonen at vælge hvad der skulle laves af mad, om testpersonen smed mindre mad ud på grund af programmet, hvor ofte de brugte programmet osv.

Formål:

Målet med denne test er at komme nærmere på en hverdagssituation hvor programmet ville blive brugt, og at se hvordan det fungerer under de forhold. Informationer om hvorvidt programmet påvirker testpersonens madspild positivt har prioritet, samt testpersonens egne tanker om deres interaktion med programmet over længere tid.

5.4 Brugertest

Som nævnt tidligere er der kun blevet benyttet tests af typen “over the shoulder”.

Vi ville gerne teste funktionaliteten samt brugervenligheden af vores program, så derfor valgte vi at lave en brugertest. Til testen anvendte vi “user-centric service development” modellen, som ses herunder. Vi kørte testpersonen igennem “user scenarios → functional → usability → evaluation”. Dvs. at der blev lavet nogle opgaver, som testpersonen skulle udføre, og ud fra testpersonens tanker evaluerede vi processen. Hensigten var at optimere programmet således at programmet bliver let anvendeligt og forståeligt for brugerne. Modellen er nyttig til formålet, og man kan i princippet køre så mange test igennem man vil.

User-centric service development

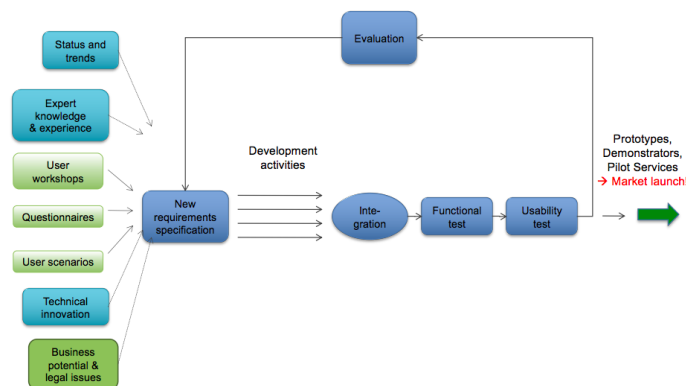


Figure 24: Den udvalgte model til kørsen af brugertest, af Henning Olesen

Vi fandt en anonym person med en ven der havde lyst til at afprøve vores

program, og bemærkede deres adfærd imens de brugte programmet. Vi gav vores testperson et sæt opgaver, der hver udforskede forskellige dele af vores program, med det formål at teste diverse funktionaliteter af vores program.

1) Indtast indkøb

Den første opgave var, at brugeren skulle forestille sig at de lige var kommet hjem fra en indkøbstur, hvor de blandt andet købte to gulerødder og nu skulle de tilføje dem til deres digitale køleskab. Brugeren navigerede gennem menuerne fint, og de bemærkede også at vores ingredienser var sorteret i alfabetisk rækkefølge. Overordnet opstod der ikke meget forvirring, på nær da brugeren skulle rette antallet af en givet ingrediens. Brugeren spurgte om man skulle skrive det antal man havde, eller rette det nuværende antal til med addition og subtraktion.

2) Lav en opskrift

Den næste opgave var, at brugeren skulle lade som om at de gerne ville lave spaghetti bolognese, og derfor skulle finde den opskrift samt bekræfte at de havde lavet den. Brugeren fandt frem til opskriften i godt tempo. Dog opstod der forvirring da brugeren modtog en besked om at de manglede en ingrediens. Brugeren fik besked om at de skulle lade som om, at de så valgte at tage ned i supermarkedet og købe den manglende ingrediens. Herefter valgte brugeren at gå ud af opskriften, og tilbage til det digitale køleskab, hvorefter de indtastede den manglende ingrediens. Brugeren troede at det ikke var muligt at bekræfte en ret, som man ikke opfyldte kravene til.

3) Fjern gammel fødevare

Den sidste opgave var, at brugeren skulle fjerne tre tomater, der havde overskredet udløbsdatoen. Dette gjorde brugeren uden problemer.

Evaluerings:

Brugertestene bekræftede vores egne erfaringer med programmet. Layout, design og mulighederne for at navigere er lette og simple. Dette er noget som vi havde sat fokus på, da en overkompliceret brugergrænseflade ville skræmme potentielle brugere væk. Brugertestene bekræftede vores mistanker om de mangler, vi selv mente at vores program har. Det drejer sig om man-

glende funktionaliteter, såsom funktioner, der kunne give brugeren mere kontrol over deres “digitale køleskab”, ved at have brugerdefinerede ingredienser i stedet for en givet mængde.

6 Diskussion, perspektivering og konklusion

6.1 Diskussion af programmet

Vores løsning til vores problemformulering kan ses som et form for “proof of concept”, altså et produkt som giver en basal løsning og som er hæmmet af manglende funktioner. En fuldkommen løsning ville indeholde flere funktioner, samt optimeringer af vores nuværende iteration.

Nogle af disse potentielle nye funktioner har vi tidligere drøftet i vores kravspecifikation og MosCow. Disse var funktioner som var placeret under “Won’t have” i vores MosCow-model, da vi manglede den tekniske viden. Derudover, så har programmeringssproget C sine egne begrænsninger der ikke gjorde det muligt for os at implementere disse funktioner. Et eksempel på disse begrænsninger kunne være en funktion, der ville gøre det muligt for brugeren at indtaste deres egne ingredienser. Programmeringssproget har ikke egenskaben til at deklarere variabler dynamisk, så vi havde ikke muligheden for at implementere denne funktion, som umiddelbart ville have bidraget meget til vores løsningsforslag og øge brugervenligheden.

En anden funktion som er værd at nævne er kvitterings-scanneren. Denne funktion ville gøre det nemmere at indsætte dagligvarer i det digitale køleskab, da man ved at scanne en kvittering kunne tilføje sine indkøbsvarer til sit digitale køleskab. Dette var en meget ambitiøs funktion, som vi ikke havde mulighed for at tilføje på grund af manglende viden og tidsbegrænsninger.

Vi havde også et ønske om at kunne scanne produktets holdbarhedsdato, formuleret i “Could have” som “udløbsdato-tjek”. Siden dette ikke var muligt, på grund af manglende erfaring i programmering og begrænsningerne i C, valgte vi ikke at udvikle videre på denne idé. Fokus på programmet i projektet har i udgangspunktet været, at det skulle være nemt at tilrettelægge og bruge for vores målgruppe. En potentiel faldgrube i brugen af et program kan være, at for meget indtastningsarbejde demotiverer brugeren, og derfor har vi så vidt muligt forsøgt at undgå for meget tastearbejde. Det kan argumenteres, at hvis brugeren selv skulle taste udløbsdatoen ind på de forskellige fødevarer, ville det hurtigt blive omsonst, og det ville umiddelbart være den eneste måde vi i vores program ville kunne registrere udløbsdatoen.

En relevant egenskab, som ikke nåede at blive tilføjet, var en form for sortering af opskriftslisten. Denne funktion ville kunne give brugeren mulighed for at prioritere sin opskrift efter egne behov. Hvis brugeren er vegetar, pescetar eller kødspiser, så ville de nemt kunne finde en opskrift der opfyldte deres behov. Denne funktion kunne endda udvides til at prioritere opskrifter alt efter de nødvendige fødevarers holdbarhed, hvis det havde været muligt automatisk at taste udløbsdatoen med. I tilfælde af at en af brugerens ingredienser var ved at udløbe, så ville denne ingrediens blive prioriteret først og brugt i en opskrift før den smides ud.

Disse funktioner ville både opfylde brugerens behov, og bidrage til at løse vores problemstilling.

6.2 Perspektivering

Vores program er kun en lille del af den samlede løsning. Madspild blandt unge voksne er et sammensat problem, der påvirkes af mange faktorer, og det gør at en fuldstændig løsning også skal tage højde for flere omstændigheder. Derudover er det klart, at vores program absolut kunne være en del af denne løsning. Fundamentet for vores program, er at man skal få opskrifter baseret på de madvarer man har til rådighed og på denne måde undgå at mad går til spilde. Dette løser den del af problemet, som handler om at man smider madvarer fordi man ikke har viden om hvordan de skal bruges.

Ser man programmet i sammenhæng med andre eksisterende løsninger, er der stadig noget der mangler for at den virkelig skal skille sig ud. For at programmet virkelig skal gøre en stor forskel og komme med noget helt nyt, må den fungere i samspil med andre produkter. Sådan som programmet fungerer i dag skal man selv indtaste hvad man har i køleskabet. Hvis dette kunne ske automatisk når man handler ind, ville det kunne gøre en endnu større forskel. Når man eksempelvis betaler med Apple Pay i dagligvarebutikker er det muligt at få en digital kvittering med oversigt over, hvad man har købt. Hvis denne kvittering automatisk blev overført til vores program, ville bruge nemmere kunne overskue sin beholdning i køkkenet, og brugervenligheden ville være optimal.

Madspild er et komplekst og sammensat problem, og den del af madspild

vores program hjælper med er kun en del af dette problem. En videreudvikling af vores program kunne være at prøve at hjælpe med mere end kun at bruge de madvarer, man har til rådighed. Hvis programmet også kunne kommet med tips angående korrekt opbevaring af mad og indkøb, ville det kunne reducere madspild yderligere.

I det videre arbejde med og udvikling af vores program er står det klart, at programmet ville fungerer bedst som applikation på en smartphone. Hvis man derudover finder en måde for app'en til at samspille med andre programmer, ville det kunne optimere brugervenligheden yderligere.

6.3 Konklusion

Efter udvikling af vores produkt kommer det endelige spørgsmål: har vi løst vores problemformulering? Som ser ud som følgende:

”Unge voksne alderen 20-34 er den målgruppe, der begår mest madspild, og det sker for det meste ved at der bliver smidt meget mad ud fra køleskabet. Dette skyldes ofte et mistet overblik over hvad de har af indkøb, og en uforudsigelig hverdag.

Hvordan kan man vha. et program, der fokusere på at brugeren skal spise indholdet i deres køleskab, hjælpe unge voksne med at få et bedre overblik og tømme køleskabet fremfor at smide mad ud? ”

Det program, der er blevet udviklet i dette projekt, løser problemformuleringen på det punkt at det hjælper unge voksne med at danne overblik over den mad de har til rådighed. Programmet hjælper aktivt brugeren til at fokusere og benytte sig af den mad, vedkommende har liggende. Derudover tilbyder programmet en nem mulighed for kun at købe den mad, brugeren har brug for, og reducerer derved madspild på sigt. Dog er løsningen ikke fuldendt. Som programmet er nu virker det kun for meget specifikke ingredienser, og har kun en meget lille mængde opskrifter det kan fremvise. Altså fungerer det mere som et “proof of concept” end et egentligt fuldt ud funktionsdygtigt program, som kan bruges i praksis. Dog vil vi mene, at vores arbejde viser at dette er en kyndig fremgangsmåde til at reducerer madspild, og at programmet i en videre udvikling ville kunne løse problemstillingen angående unge voksnes madspild på en mere fyldestgørende måde.

7 Evaluering

Efter gennemførelse af dette projekt er der adskillige vigtige pointer, som kan tages videre. Det er blevet gjort klart at unge voksne, på trods deres store engagement i miljøet, er den aldersgruppe der smider mest mad ud. Derudover er vi som gruppe blevet bekendt med hvor mange begrænsninger programmeringssproget C har, når det gælder skabelsen af en funktionel GUI som brugeren kan interagere med.

Hvis dette skulle bringes videre og ud af “proof of concept”-stadiet, ville vi foreslå at arbejde i et programmeringssprog som ikke er C, da dette ville give mulighed for at lave en ordentlig GUI, som markant ville forbedre brugerens interaktion med programmet. Derudover så ville implementeringen af nogle af de funktioner, som vi diskuterede men ikke fik mulighed for at implementere, også forbedre programmet. Specielt ville en liste af ingredienser, som ikke var fast, og en markant større liste af opskrifter være til stor gavn. Alt i alt kan det program som blev produceret virke som et godt fundament til et endeligt program.

Bibliografi

- [1] L. o. F. Ministeriet for Fødevarer. “Første samlede kortlægning af madspild i danmark offentliggøres.” (), [Online]. Available: <https://fvm.dk/nyheder/nyhed/nyhed/foerste-samlede-kortlaegning-af-madspild-i-danmark-offentliggoeres>.
- [2] D. Statistik. “Befolkningstal.” (), [Online]. Available: <https://www.dst.dk/da/Statistik/emner/borgere/befolkning/befolkningstal>.
- [3] C. Petersen, *Kortlægning af madaffald i servicesektoren: detailhandel, restauranter og storkøkkener*. Miljøstyrelsen, 2014.
- [4] L. o. F. Ministeriet for Fødevarer. “Madspild.” (), [Online]. Available: <https://www.foedevarestyrelsen.dk/Leksikon/Sider/Madspild.aspx>.
- [5] L. og Fødevarer. “Mad og madspild.” (), [Online]. Available: https://mindthetrash.dk/wp-content/uploads/sites/2/MadogMadspild_elev_aktivitet1.pdf.
- [6] M. Minter. “Madspild er klimabelastning til ingen verdens nytte.” (), [Online]. Available: <https://concito.dk/nyheder/madspild-er-klimabelastning-til-ingen-verdens-nytte>.
- [7] E. Commission. “Carbon footprint of food waste not necessarily related to its weight.” (), [Online]. Available: https://ec.europa.eu/environment/integration/research/newsalert/pdf/carbon_footprint_of%5C%20food_waste_not_necessarily_related%5C%20_to_its_weight_406na3_en.pdf.

- [8] T. G. to Go. "Konsekvenserne ved madspild." (), [Online]. Available: <https://toogoodtogo.dk/da/movement/knowledge/the-carbon-footprint>.
- [9] U. Nations. "Goal 12: Ensure sustainable consumption and production patterns." (), [Online]. Available: <https://www.un.org/sustainabledevelopment/sustainable-consumption-production/>.
- [10] H. Ritchie. "Food waste is responsible for 6% of global greenhouse gas emissions." (), [Online]. Available: <https://ourworldindata.org/food-waste-emissions>.
- [11] V. Stancu and L. L. hteenm
ki, *Consumer food waste in Denmark*. DCA-Nationalt Center for Fødevarer og Jordbrug, 2018.
- [12] B. N. Cathrine Bloch og Laura Juul. "Gymnasieelever og studerende stresser helt vildt." (), [Online]. Available: <https://www.berlingske.dk/samfund/gymnasieelever-og-studerende-stresser-helt-vildt>.
- [13] B. T. Hartkopf. "Hver tredje nye studerende føler sig meget stresset." (), [Online]. Available: <https://www.eva.dk/videregaende-uddannelse/tredje-nye-studerende-foeler-sig-stresset>.
- [14] R. K. Wulff. "Unge indkøbs- og madvaner." (), [Online]. Available: <https://www.madkulturen.dk/wp-content/uploads/2021/05/Unge-indkoebs-og-madvaner-Mindre-madspild-blandt-unge-Maj-2021.pdf>.

- [15] S. Suranga. “Why every developer should start programming with c.” (), [Online]. Available: <https://betterprogramming.pub/why-every-developer-should-start-programming-with-c-39b3a87392bf>.
- [16] N. Ebdrub. “Sådan bruger du videnskabsteori.” (), [Online]. Available: <https://videnskab.dk/miljo-naturvidenskab/sadan-bruger-du-videnskabsteori>.
- [17] M. B. Madsen, “Klimabarometeret 2020.” [Online]. Available: https://concito.dk/sites/concito.dk/files/media/document/Klimabarometeret%5C%202020_f%5C%C3%5C%A6rdigrapport.pdf.
- [18] D. Statistik. “Hver sjette reducerer madspild ved brug af apps.” (), [Online]. Available: <https://www.dst.dk/da/Statistik/nyt/NytHtml?cid=39771>.
- [19] Studentnum. “Årets studieby: Om undersøgelsen.” (), [Online]. Available: <https://www.studentum.dk/aarets-studieby/arrets-studieby-metode-18584>.
- [20] Bolius. “Studieboliger og værelser er billigst i aalborg.” (), [Online]. Available: <https://www.bolius.dk/studieboliger-og-vaerelser-er-billigst-i-aalborg-29511>.
- [21] I. D. Foundation. “User scenarios.” (), [Online]. Available: <https://www.interaction-design.org/literature/topics/user-scenarios>.