
P3: Noget med indkøb

- Programmering og modelløsning -

Projektrapport
DS304E14

Mathias Corlin Mikkelsen

Marc Tom Thorgersen

Morten Pedersen

Mathias Sass Michno

Søren Hvidberg Frandsen

Troels Beck Krøgh

Aalborg Universitet
Institut for datalogi
Selma Lagerlöfs Vej 300
9220 Aalborg Ø



AALBORG UNIVERSITET
STUDENTERRAPPORT

3. Semester — Software

Selma Lagerlöfs Vej 300

9210 Aalborg SØ

<http://www.cs.aau.dk/>

Titel:

Noget med indkøb

Tema:

Programering og modelløsning.

Projektperiode:

P3 (Efterårssemestret 2014)

02-09-2014 - 19-12-2014

Projektgruppe:

DS304E14

Gruppemedlemmer:

Marc Tom Thorgersen

Søren Hvidberg Frandsen

Troels Beck Krøgh

Mathias Corlin Mikkelsen

Morten Pedersen

Mathias Sass Michno

Abstract:

The following project studies how to fuk

Vejleder:

Thomas Bøgholm

Oplagstal: x

Rapport sideantal: x

Appendiks sideantal: x

Total sideantal: 48

Projekt klaret den:

19-12-2014

Ord/Tegn (Cirka): ...

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.

Forord

Her vil komme tekst om læsevejledning.

Teksten vil beskrive kildehenvisninger, samt kodeeksempler.

Ord som 'vi' referer til forfatterne af rapporten.

Eventuel beskrivelse af struktur på: CD, bilag og andet.

Indholdsfortegnelse

A	Analyse	1
Kapitel 1	Indledning	3
Kapitel 2	Problemanalyse	5
2.1	Indledende interview	5
2.2	State of the art	6
2.3	Prototype interviews	11
2.4	Problemformulering	12
Kapitel 3	Systemanalyse	13
3.1	Systemdefinition	13
3.2	Modellering af System	14
3.3	Problemområde	15
3.4	Analyse af anvendelsesområdet	18
3.5	Kravspecifikation	21
B	Udvikling	23
Kapitel 4	Udvikling	25
4.1	MVC-mønsteret	25
4.2	Klassediagram	26
4.3	eTilbudssavis API	29
C	Refleksion	33
Kapitel 5	Data-Flow diagram	35
Kapitel 6	Konceptuelle scenarier	37
D	Referencer	38
	Litteratur	39
	Figurer	40
	Tabeller	40

Del A

Analyse

Indledning

1

I dagligdagen kan det være svært at koordinere indkøb, opskrifter og tilbud således, at dette foregår på en optimal måde. Nogle kigger i tilbudsaviser og planlægger deres indkøb i en eller flere butikker, andre vælger kun at handle i en butik. Hvis ikke man planlægger disse indkøbsture, kan det lede til madspild pga. overindkøb/impulskøb eller for mange unødige ture i supermarkederne, hvilket kan tage tid i ens hverdag. Derudover kan det være problematisk at koordinere, hvem der handler hvad ind med eventuelle samboende.

Det kan også være svært at leve sundt og varieret som fødevareministeriet anbefaler; det kan være svært at finde på noget at lave til aftensmad; eller også har man ikke nogen gode opskrifter, man synes om. Hvis man planlægger sine indkøb efter opskrifter, skal man holde styr på, hvilke varer man har i sit hjem for at undgå at købe for meget ind. Det kan samtidig være svært at overskue, hvilken butik der er den bedste at handle ind i på et givent tidspunkt, eftersom at tilbud hele tiden ændrer sig og faste priser er forskellige. Det er heller ikke sikkert, at den valgte opskrift man køber ind efter, indeholder varer, som er på tilbud, når man gerne vil have dem - på denne måde ender man måske med at leve dyrere end nødvendigt.

Dette projekt vil undersøge, om man kan gøre det nemmere at handle efter tilbuddene i dagligvarebutikkerne, alt imens man reducerer problemerne vedr. det at beslutte sig for, hvad man skal lave til aftensmad. Derfor vil der i de følgende afsnit blive udforsket hvad der er på markedet til hjælp med denne problemstilling. Desuden vil der også undersøges personers vaner, samt holdninger til elektronisk hjælp til problemstillingen i form af korte interviews.

Problemanalyse 2

I dette kapitel lægges rammerne for problemet fast. Problemet som udforskes her, er problematikker i forhold til madlavning og indkøb i hverdagen.

Først foretages en indledende interviewrunde der har til formål at bekræfte og klarlægge problemet, samt at undersøge om der er interesse for løsninger til problemet. Efter dette foretages en state of the art-analyse, der har til formål at undersøge og udforske eksisterende løsninger til problemet. Der skal altså undersøges hvilke funktionaliteter løsningerne har, samt om løsningerne er tilfredsstillende. Efter denne undersøgelse, udføres endnu et interview. Dette interview skal undersøge hvilke funktionaliteter potentielle brugere har interesse for, hvilket gøres i forbindelse med gruppens egen prototype. Sidst i kapitlet defineres problemformuleringen, som danner grundlag for den resterende del af projektet.

2.1 Indledende interview

I dette afsnit præsenteres resultaterne af den første interviewrunde, som blev foretaget som start på projektet. Formålet med denne interviewrunde er at bekræfte, at en datologisk løsning kan afhjælpe eller minimere problemet, som vi her analyserer. Interviewene har også til formål at undersøge hvilke problemer personer har med madlavning og indkøb dertil.

Onsdag d. 17 september gennemførte vi en række semi-strukturerede interviews i Aalborg midtby udenfor Føtex. Vi adspurgte syv personer i forskellige målgrupper, herunder unge der bor alene, unge der bor sammen med nogen, og folk over 30 med begge førnævnte sociale statusser. Aldersspændet var fra 20 - 51, og kønnene var ligeligt fordelt. Formålet med denne interviewrunde var at danne et basalt overblik over vores problemområde, samt undersøge, hvordan folk håndterede vores initierende problem på nuværende tidspunkt. Vores interviews bestod af en håndfuld forberedte spørgsmål, samt løsere samtale, for at følge op på disse spørgsmål. Spørgsmålene findes på ¹ Ud fra denne interviewrunde er der gjort følgende observationer:

- Generelt er folk uorganiserede når det kommer til indkøb, og især unge mennesker bruger sjældent indkøbslister.
- Inspiration til aftensmåltider findes ofte i indkøbssituationen eller ud fra princippet "hvad man lige har lyst til".

¹FiXme Note: indsæt link til interviewguiden her med de tilhørende spørgsmål.

- Det er ikke mange, der gennemgår tilbudsaviser, og endnu færre går bevidst efter tilbud. Dog kan det påvirke beslutninger om aftensmåltid, hvis en given varer er på tilbud i butikken, hvor der handles ind.
- Især den yngre målgruppe (18-28) er interesseret i en løsning med tilbud, indkøbslister og opskrifter. Mens den ældre målgruppe (28+) virker mere interesseret i at kunne dele indkøbslister med familie, for bedre at kunne organisere indkøb.
- Opskrifter er som sagt noget, der mest interesserer det yngre segment. Dog påpeger nogle, at stor tendens til at være kræsen kræver at man kan sætte præferencer.

Det viste sig derfor at der er interesse for en løsning der kan hjælpe med inspiration til madlavningen. Tilbud havde en effekt på hvad man valgte at spise, og derfor kan tilbud være en central del af en sådan løsning. Der var ikke mange af de interviewede der brugte løsninger som allerede var på markedet, om dette skyldes mangel på viden om dem, vides ikke. Vi vil derfor undersøge hvilke løsninger der er på markedet i det næste afsnit som hjælper med følgende emner:

- Indkøbslister
- Tilbud
- Opskrifter.

2.2 State of the art

I dette afsnit har vi forsøgt at finde en bred vifte af forskellige løsninger der kan løse et eller flere problemer i forbindelse med indkøb og madlavning, omhandlende indkøbslister, tilbud og opskrifter.

Der vil blive kigget på web services og applikationer, der har til formål at løse nogle af de samme problemer, der berører vores emne. Løsningerne, der er præsenteret i dette afsnit, forsøger at afhjælpe de problemer beskrevet af de adspurgte i interviewene i afsnit 2.1 på forrige side.

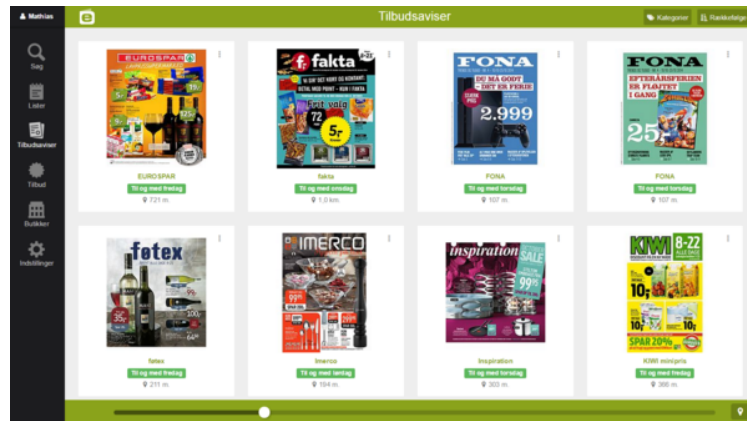
2.2.1 eTilbudsavis

eTilbudsavis er en online service, der kan findes på deres hjemmeside². eTilbudsavis er en online avis med en høj funktionalitet og en nem brugergrænseflade. Der kan på siden oprettes et login, således der kan findes tilbage til ændringer på et senere tidspunkt eller andre enheder. eTilbudsavis har tre mærkbare funktioner, som brugeren har adgang til, hvilket er tilbudsaviser, ønske- og indkøbslister og tilbud.

Tilbudsavis-funktionen, hvilket kan ses på figur 2.1 på næste side, indeholder også den mulighed at sætte en præference på, hvor store en radius man vil lede i efter butikkers tilbudsaviser. Under hver avis, som det ses på figuren, er der også angivet afstand i meter fra ens aktuelle placering. Der kan vælges imellem alle aviser, som er tilgængelige online og inden for den valgte radius. Aviserne bliver opdateret løbende, så når der er en ny tilgængelig, bliver den gamle fjernet. Inde i aviserne kan man trykke på en vare, og den bliver tilføjet til en liste.

²www.etilbudsavis.dk

Der er mulighed for at tilføje varer til to lister, en indkøbsliste og en ønskeliste. Når en vare er valgt, bliver den tilføjet til den valgte liste. Listen indeholder navn, butik, pris og den valgte mængde, for varen. Der kan foruden det at vælge varer fra tilbudsaviserne, nemt skrives generiske varer på listen. Varen på listen kan da krydses af for at kunne holde styr på, hvad der er blevet købt.



Figur 2.1. Tilbudsavis på eTilbudsavis.dk

Hvis der ikke ønskes at skulle bladres tilbudsavis igennem, er der den mulighed at få vist en hel side kun med tilbud. Alle aktuelle tilbud fra aviserne, er da vist som elementer med navn, beskrivelse, pris, butik og afstand. Disse tilbud kan på samme måde nemt tilføjes til listerne.

eTilbudsavis er en god online løsning, som har mange gode features, derfor har vi også brugt dem som kilde til vores tilbud gennem deres API. Dette API er beskrevet i afsnit 4.3 på side 29.

2.2.2 Tilbudsgugen

Tilbudsgugen minder på mange måder om eTilbudsavis, og kan findes på deres hjemmeside³. Den har samtlige dagligvareaviser, samt flere inden for bl.a. byggemarkeder, og autoudstyr. De giver et nemt overblik over diverse aviser, og man kan hurtigt og nemt læse dem på nettet. Der er desuden mulighed for at lave præferencer som ved eTilbudsavis, her kan man bl.a. vælge økologi eller nøglehulsmærket. Når man tilføjer en vare til indkøbslisten, søger den automatisk efter tilbud på den valgte vare. Man bliver bedt om at vælge et specifikt tilbud, og netop dette tilbud bliver tilføjet til indkøbslisten med pris, butik, udløbsdato, mængde og et billede af varen. Man kan som i eTilbudsavis også trykke på en vare direkte i avisen for at tilføje den til sin liste. Hvis man vil dele sin indkøbsliste, er det også en mulighed vha. en "delekode" som man kan give til en anden bruger - de kan på denne måde også se listen. Funktionerne findes på hjemmesiden, men det er ikke altid de virker. For eksempel hvis man tilføjer noget uden at angive et antal, og du så prøver at dele listen med en, vil de ikke være at finde på listen. Desuden kan man ikke ændre på antallet af varen, du allerede har sat på din indkøbsliste. For at opnå dette, skal man slette varen, og tilføje den igen med det nye antal. De har desuden også en smartphone app, men denne crasher ofte, når man benytter sig af deres indkøbsliste, men fungerer tilgængelig fint, hvis man blot vil se på ugens tilbud i aviserne.

Tilbudsgugen er et lidt dårligere alternativt til eTilbudsavis, da der er problemer med delbarheden af indkøbslisterne, samt den app, der er stillet til rådighed, ikke er stabil. Derimod er den nem at navigere, siden er brugervenlig og ser overskuelig ud.

³www.tilbudsgugen.dk

2.2.3 Smartphone Apps udgivet af butikker

På tabel 2.1 ses der nogle butikskæder, som har udviklet android apps, samt hvilke funktioner disse har. For at skabe et overblik, er der udvalgt features, og disse er blevet opsat i et skema for overskuelighedens skyld.

	Tilbudssavis	Indkøbsliste	Opskrifter	Varescan	Find butik	Budget	Madplan	Rabatkupon	Deling	Play rating	Senest opdateret
Føtex	✓	✓	✓	✓	✓					3.4 (354)	2014-07-24
SPAR	✓	✓	✓		✓	✓				2.8 (64)	2014-05-15
Fakta	✓	✓	✓		✓		✓	✓	✓	3.1 (454)	2014-08-02
FaktaQ	✓		✓		✓					4.4 (7)	2014-03-11
REMA 1000	✓	✓	✓		✓					3.5 (674)	2014-04-16
SuperBrugsen	✓	✓	✓		✓					3.8 (987)	2014-06-30
Kvickly	✓	✓	✓		✓					3.7 (632)	2014-07-20

Tabel 2.1. Nogle butikskæder med smartphone apps samt deres funktionaliteter.

Vi har valgt at undersøge en af disse apps nærmere, og valget faldt på Faktas app, MitFakta, da denne bliver opdateret og har flest funktioner.

Faktas Android App



Figur 2.2. MitFakta

Faktas android app kalder de for “Mit fakta”, den er som helhed overskuelig. Efter åbning af appen kan man vælge mellem seks menupunkter eller ændre sine indstillinger, disse kan ses på figur 2.2. Første menupunkt omhandler “Coop-kortet”, og giver brugeren mulighed for at indtaste sine medlemsinformationer, da disse giver særlige tilbud. Andet menupunkt er deres tilbudssavis, hvilket er en digital kopi af den fysiske tilbudssavis. Dog kan man fra den også tilføje varer til sin indkøbsliste, eller se varerne i et gitterformat. Tredje menupunkt er “indkøbsliste”, her kan man have flere personlige og/eller delte indkøbslister. Her kommer deres Facebook integration også i spil, som tillader nem deling af indkøbslister med brugerens Facebook venner. Fjerde menupunkt er en madplan, hvori man kan planlægge sin egen madplan, eller se Faktas anbefalinger til en “under 20 kr pr. person pr. aften”-løsning. Femte menupunkt er deres opskrifter. Her findes der et stort antal af opskrifter, disse kan tilføjes direkte til ens madplan eller indkøbslister (både personlige og delte). Sjette menupunkt hedder “Åbningstider”, og her kan brugeren finde Faktas butikker samt deres forskellige åbningstider.

Appen virker rigtigt godt, der er ingen blindgyder, og derfor er den meget nem at navigere rundt i. Der er lagt meget fokus på den sociale del, da der kan deles med venner på facebook, hvilket simplificerer delingen af indkøbslister. Den eneste ulempe, der er ved denne app, er, at den kun er egnet til vare fra Fakta, og derfor sætter en del restriktioner på sig selv.

2.2.4 Tøm køleskabet

Der findes talrige tjenester, der tilbyder "at tømme dit køleskab". Mere specifikt, tilbyder de en service, hvor du som bruger, angiver hvilke varer dit køleskab pt. indeholder, samt hvilke andre ingredienser du har til rådighed. Derefter får du så præsenteret en række forskellige opskrifter, der kan laves ud fra dine tilgængelige ingredienser. De fleste af tjenesterne (herunder dem vi her har undersøgt) viser også opskrifter, som indeholder yderligere ingredienser. Dette betyder naturligvis, at man som bruger ikke bliver fritaget fra at handle ind, hvis man mangler nogle ingredienser til lige netop den opskrift, man vælger at udføre. Tjenesten *MyFridgeFood*⁴ tilbyder at oprette en indkøbsliste ud fra netop disse manglende ingredienser, hvilket kan lade sig gøre blandt andet fordi, alle opskrifter er interne på *MyFridgeFood*. I modsætning til dette er der *Supercook*⁵, som linker til eksterne opskrifter, og ikke tilbyder at generere en indkøbsliste. Umiddelbart anbefales opskrifter ikke ud fra den enkelte brugers smag og madvaner, men udelukkende på baggrund af, hvad man "har i køleskabet". Grundet dette virker tjenesterne mere som simple filtreringer af databaseopslag, end egentlige anbefalinger, der tager højde for brugerens smag og præferencer indenfor den gastronomiske verden.

2.2.5 Alternativer til Indkøbsturen

Der findes altså allerede mange applikationer, i form af apps til smartphones og webapps, som kan hjælpe med indkøbsturen og planlægning af måltider. Hvis disse indeholder opskrifter, som tager højde for tilbudene i en given uge, er det inden for en enkelt butikskædes tilbud. Der er altså ingen applikationer, der tager højde for tilbud fra alle butikskæderne. I dette afsnit undersøges alternativer til indkøbsturen.

Ud over mobilapplikationer, flytter nogle dagligvarebutikker på internettet, sådan man kan handle ind hjemmefra, som man kender det fra almindelig nethandel. Andre løsninger er baseret på abonnementsordninger, som fjerner meget af planlægningen fra ordningens bruger. Herunder gennemgås eksempler på nogle firmaer, der tilbyder de to ovennævnte løsninger.

De seneste år har online dagligvare-butikker skudt frem, og selvom de er i kraftig vækst udgør de kun en lille del af det samlede dagligvareforbrug i Danmark[4].

Online butikker

Eksempler på online dagligvarebutikker er SuperBest⁶ og Irma⁷. Nogle af butikkerne har dog kun en online butik, og ingen fysisk modpart, dette er butikker som Nemlig.com⁸ og Osuma⁹. Disse butikker kan gøre det lettere at handle ind. Sådanne butikker giver både mulighed for at

⁴www.myfridgefood.com

⁵www.supercook.com

⁶www.superbest.dk

⁷www.irma.dk

⁸www.nemlig.com

⁹www.Osuma.dk

handle ind hjemmefra, fra studiet eller fra arbejdet i en pause. Ved at handle hjemmefra kan man også let overskue, hvad man mangler, imens man handler ind, og man kan derfor undgå at glemme at skrive noget på indkøbssedlen. Ligeledes undgår man at stå i supermarkedet og være i tvivl om, man mangler en bestemt vare. Nogle af firmaerne tilbyder også opskrifter. Nemlig.com tilbyder opskrifter, og hjemmesiden kan præsentere alle deres varer, som indgår i listen over ingredienser. Dette gør, at man direkte fra en given opskrift, kan sammenligne tilsvarende varer og tilføje en af dem til sin indkøbskurv. Andre butikker giver også mulighed for at oprette madplaner, hvilket kan gøre planlægningen af indkøb og madlavning i hjemmet lettere. Disse online butikker kommer dog også med forskellige ulemper i forhold til fysiske supermarkeder. Det koster penge at få leveret sine varer fra disse butikker, typisk omkring 50 kr., og i nogle af butikkerne skal man købe for et minimumsbeløb. Ved handel gennem online butikkerne, kan du ikke få varerne med det samme; det varierer butikkerne imellem, hvor stor ventetiden er. Disse to ulemper gør ligeledes, at man ikke kan benytte disse løsninger, hvis man står og mangler en vare eller to.

Abonnementsordninger

Ordningerne sender typisk kasser med daglivarer ud til deres kunder. Disse kassers indhold tilbyder ofte en varieret blanding af kød, grønt og frugt. I denne kategori findes udbydere som Aarstiderne og igen SuperBest[1, 7]. Ideen med sådanne kasser er, at nogle ansatte, typisk kokke eller lignende ved eksempelvis Aarstiderne, har sammensat kasser med ingredienser til mellem tre og fem aftensmåltider. På deres hjemmeside eller mobilappikation kan man så finde guider og opskrifter til, hvordan maden tilberedes. Derudover har Aarstiderne også suppleret med ydeligere tre kategorier, for at de kan tilbyde mere fleksible løsninger til kunderne. Den første kategori holder sig inde for kasse-dogmet og er generelle kasser, hvor man eksempelvis kan købe en kødkasse, fiskekasse eller en frugtkasse; i disse kategorier findes der så forskellige kasser inden for hver kategori. Anden kategori er suppleringskasser, hvorigennem kunderne kan købe forskellige varer i løssalg, hvis ikke kasserne opfylder deres behov. Sidste kategori er morgenmad, her tilbyder de en blanding mellem kasser og løssalg. Denne model kan løse mange problemer i forhold til madlavningen for kunderne. Løsningen vil kunne mindske den planlægning en person behøver foretage sig, fordi kasserne indeholder alle ingredienser til flere måltider, samt opskrifter at følge. Dette betyder, at kunden hverken behøver planlægge aftensmaden eller indkøbet dertil. Ligeledes ville den kraftigt mindske behovet for at handle ind andre steder. Denne løsning kommer dog heller ikke uden ulemper. Dogmet med at forudbestemme flere måltider og have de eksakte ingredienser, begrænser den kreativitet, som en person normalt ville kunne udleve i køkkenet. Ydermere begrænses ideen meget af den moderne families struktur, hvor det ofte ikke er hele familien, der spiser sammen mange dage om ugen. Det er også svært at tage højde for præferencer som kræsenhed eller allergier i disse forudbestemte kasser. Denne løsning vil heller ikke kunne fritage en bruger fra at skulle handle ind andre steder end her, da de ikke tilbyder alt.

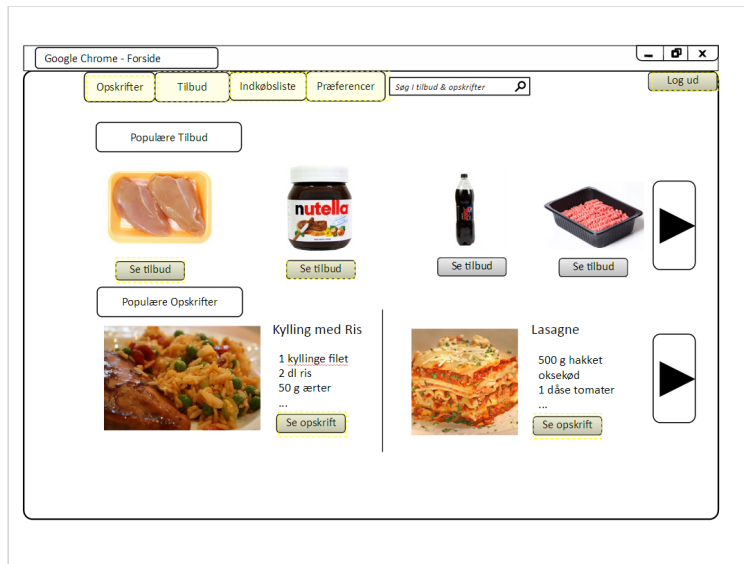
Som vi har set i ovenstående gennemgang, findes der alternativer til den klassiske indkøbstur, der er dog ingen af disse, der formår at løse alle problemerne. Ved alle løsningerne opstår også nye ulemper og problemstillinger i forhold til almindeligt indkøb. Alt taget i betragtning vil det komme meget an på nuværende vaner og familiestrukturen, om disse muligheder vil være en god løsning for en given familie.

2.3 Prototype interviews

State of The Art undersøgelsen viste at der er mange forskellige løsninger, men der er også mange mangler i disse alligevel. Ingen af løsningerne tager tilbud fra alle butikker, med en indkøbsliste, opskrifter med tilbudene, samt vurdering af disse. Derfor har vi udformet en prototype i Microsoft Office Powerpoint som viser forskellige funktionalteter. Diasshowet består af forskellige skærbilleder, som der kan navigeres imellem ved tryk på de gule felter.

Et eksempel kan ses på figur 2.3.

Prototypen bruges i vores 2. runde af interviews, som har til formål at forstå hvilke funktionalteter potentielle brugere ønsker i en løsning. De interviewede bestod mest af unge personer i starten af tyverne, hvor af de fleste var studerende. Dette valg blev taget da ud fra det første interview, blev det konkluderet at denne gruppe havde størst interesse i en løsning. Interviewets fremgangsmåde kan findes



Figur 2.3. Forside fra Prototypen, der blev brugt i forbindelse med vores 2. runde af interviews

FiXme Note! på ¹⁰

Ud fra denne undersøgelse står det nu klar at en løsning skal have følgende funktionalteter:

- Indkøbsliste integreret med tilbud.
- Oversigt over tilbud fra samtlige dagligvarebutikker.
- Opskrifter, som gør brug af tilbud.
- Mulighed for at vurdere opskrifterne og få anbefalet lignende.
- Valg af hvor man vil handle, hvilke madvarer man ikke vil spise osv.
- Deling af indkøbsliste med andre.
- Filtrering under opskrifter, så man kan se hvad man kan lave ud af f.eks. oksekød osv.
- Overvågning af varer, så man får en form for besked hvis noget kommer på tilbud en uge.

Der var også andre funktionalteter som nogle af de adspurgte fandt relevante. F.eks. blev en kalorietæller foreslået af en af de adspurgte, men dette vælges der at afgrænses fra, da det ikke hænger så godt sammen med resten af den meget tilbudsorienterede løsning.

¹⁰FiXme Note: Indsæt kilde til interview fremgangsmåde.

Herudover gav de interviewende også forslag og råd angående hvilke funktionaliteter skulle være hvor i et desktop layout. Disse forslag tages med i overvejelserne når brugergrænsefladen skal designes.¹¹

FiXme Note!

2.4 Problemformulering

State of the art undersøgelserne viste at der findes mange applikationer og løsninger, til at hjælpe med den daglige madlavning og indkøb til hjemmet. Det viste sig desuden at selvom der var mange løsninger, var disse ikke alle optimale, nogle grundet manglende funktionalitet, andre grundet dårlig udførelse af programmet. Vores første interviewrunde viste at der ikke blev benyttet mange hjælpemidler i forvejen, men at der dog var en interesse for emnet. Dette kan bl.a. skyldes at de allerede eksisterende løsninger, ikke har interesse for personerne, dårlig kvalitet af programmer, eller måske bare dårlig og/eller manglede markedsføring.

Anden interviewrunde viste hvilke funktionaliteter en løsning skulle have. Med disse informationer arbejdes der videre med følgende problemstilling:

FiXme Note!

12

Hvordan kan dagligvarebutikkernes tilbud gøres mere overskuelige med integrerede indkøbslister, således det er lettere at leve efter billige og varierede opskrifter til aftensmad?

¹¹FiXme Note: Indsæt evt. sætningen: En oversigt over alle de interviewedes svar kan findes på bla bla bla i appendix. - Søren

¹²FiXme Note: Passer denne problemformulering stadig ind i det nu ? Vi har jo fundet funktionaliteterne, passe den så stadig ind i det hele ? - Søren

Systemanalyse 3

Systemanalysen har til formål at finde frem til hvilke procedurer og data der skal bruges til at udforme en løsning af problemet. Først præsenteres systemdefinitionen, som har udgangspunkt i problemformuleringen og de funktionaliteter der blev opstillet i afsnit 2.3 på side 11. Efterfølgende vil der være henholdsvis en analyse af problemområdet og anvendelsesområdet, som sidst i kapitlet vil danne baggrund for en kravspecifikation.

3.1 Systemdefinition

På baggrund af kapitel 2 udarbejdes en BATOFF-analyse, som beskrevet i OOA&D[3] og efterfølgende formuleres en skreven systemdefinition baseret på disse kriterier.

Denne analyse har til formål at definere retningen for det videre arbejde i projektet. Systemdefinitionen er en kort tekst der har til formål at beskrive systemets overordnede krav og funktionaliteter.

Nedenfor ses BATOFF udarbejdelsen og efterfølgende systemdefinitionen. BATOFF kriterierne hjælper med at danne et overblik over diverse emner som en systemdefinition bør omfatte.

3.1.1 BATOFF

Betingelser

- Adgang til tilbud
- Interesse for indkøb, opskrifter og tilbud

Anvendelsesområde

- Tilbud
- Brugere
- Servere
- Klienter
- Overvågning af tilbud og madvaner
- Midler til lagring af data
- Styring af anbefalinger

Teknologi

- Smartphone (Mobil web-device)
- Tablets
- Til udvikling: Computer m/udviklerværktøjer
- Browser m/internetadgang

Objekter

- Opskrifter
- Indkøbsvarer (tilbud)
- Brugere
- Vurdering
- Præferencer
- Indkøbsliste
- Tilbudsaviser

Funktioner

- Overvågning af varer
- Håndtering af indkøbslister
- Bedømmelse af opskrifter

Filosofi

- Indkøbsassistent og inspirationsgenerering

De ovenstående kriterier indgår altså i den følgende systemdefinition:

3.1.2 Systemdefinition

Systemet hjælper på problemer, der kan opstå i forbindelse med indkøb og madlavning i hjemmet. Systemet organiserer indkøbslister, opskrifter og aktuelle tilbudsvarer, samt anbefaler varer og/eller opskrifter, baseret på tidligere valg, bedømmelser af opskrifter og præferencer ang. madvarer. Aktuelle tilbud hentes fra internettet, og kan tilføjes, sammen med generiske varer, til indkøbslister. Desuden kan det overvåge hvornår, en valgt generisk vare kommer på tilbud. Systemet tilgås via en webbrowser, således det kan bruges på både computer, tablet og smartphone. Systemet udvikles som en serverside-applikation med adgang til databaser til håndtering af system- og brugerdata. Udviklingen af systemet kræver computere med de relevante udviklingsværktøjer.

Denne systemdefinition vil nu være udgangspunkt for vores videre arbejde i rapporten.

3.2 Modellering af System

Der to komplementære synsvinkler for den model, som beskriver systemet. Dette er hhv. at systemet modellerer noget hvilket er problemområdet, samt at det anvendes af brugerne, anvendelsesområdet. Disse er defineret som følger:

Problemområde: *"Den del af omgivelserne, der administreres, overvåges eller styres ved hjælp af et system"*

Anvendelsesområde: *"En organisation, der administrerer, overvåger eller styrer et problemområde"*[3]

3.2.1 Problemområdet

Problemområdet bruges som en modellering af et problem fra den virkelige verden, hvor et system skal benyttes for at administrere, overvåge eller styre et område. Dette gøres ved at beskrive diverse klasser, som vil indgå i systemet, ud fra disse ses på hvilke hændelser, som er involveret i klasserne. Ydermere ses der på hvilken adfærd der er mellem diverse klasser, hændelser og objekter i systemet, dette giver en beskrivelse af hvilken opførsel og struktur problemområdet skal modellere. I dette projekt omfatter problemområdet planlægning af indkøb, inspiration til mad såvel som at spare penge. For at beskrive dette nærmere, ses der på klassesdiagrammer og hændelsestabeller, for at danne overblik over systemet og ende ud med en sammenhængende model for problemområdet.

3.2.2 Anvendelsesområdet

Hvor problemområdet beskriver systemet, beskriver anvendelsesområdet hvordan systemet skal anvendes. Ud fra dette spørgsmål opstilles en række af krav for systemets funktioner og grænseflade. Til dette formål ses der på brugen af systemet, hvilke typer af brugere er der, hvilke brugsmønstre er der for de individuelle funktionaliteter i systemet, samt hvordan skal funktionaliteterne tilgås fra grænsefladen. I dette projekt indebære anvendelsesområdet at kunne administrere sine indkøb, få en let oversigt over tilbud, inspiration til mad i form af opskrifter, at kunne overvåge vare, som man har interesse for at købe på tilbud.

3.3 Problemområde

Ud fra systemdefinitionen ved vi at systemet skal holde styr på følgende:

- Tilbud
- Varer
- Opskrifter

Med disse informationer kan systemet hjælpe brugeren til at finde billige varer i bestemte butikker, og eventuelt anbefale opskrifter der bruger disse tilbudsvarer. I de følgende afsnit vil disse emner blive beskrevet vha. klassebeskrivelser, en hændelsestabel, og et klassesdiagram.

3.3.1 Klasser

For at kunne håndtere de 3 allerede nævnte klasser, tilbud, varer og opskrifter, skal der være flere for at danne en sammenhæng. Denne sammenhæng vil analyseres her.

Vare: En vare indgår i opskrifter, og indkøbslister. Når man laver sin indkøbsliste vælger man hvilke varer man vil købe og skriver dem derpå. Desuden kan en vare have et antal tilbud hver uge, hvilket betyder at der også skal laves en tilbuds klasse.

Tilbud: Hver uge kommer der nye tilbud fra de fleste danske dagligvarebutikker. Disse modeleres og kobles på varen og dermed dannes der en kobling fra varen til tilbudene.

Opskrift: En opskrift har en liste over ingredienser, hvilket altså er varer. I interviewene i afsnit 2.3 på side 11, blev det nævnt at brugerne gerne ville kunne vurdere en opskrift, og dermed få anbefalet yderligere opskrifter som minder om denne. Dette leder til at der laves en såkaldt vurderingsklasse.

Vurdering: Vurderinger bliver foretaget når en bruger af systemet har lavet en opskrift, og vil give den en vurdering, både til at hjælpe andre, men også for at få lignende opskrifter anbefalet, hvis man nu synes den var god.

[Anbefaling:] En anbefaling bliver givet til en person når der er givet vurderinger, således der kan laves en anbefaling af en opskrift til personen.

Person: Vurderingerne skal kobles på personen som vurdere opskriften. Derfor laves der en person klasse, som samtidig indholder attributter som gør det muligt at sætte personens præferencer. Det er desuden også personerne som har deres egne indkøbslister.

Indkøbsliste: Indkøbslisten laves af personen, og fyldes op med objekter fra vare klassen. Indkøbslisterne kan være delte, og derfor skal koblingen på personen klassen ikke være singulær, men altså 1..*.

3.3.2 Hændelser

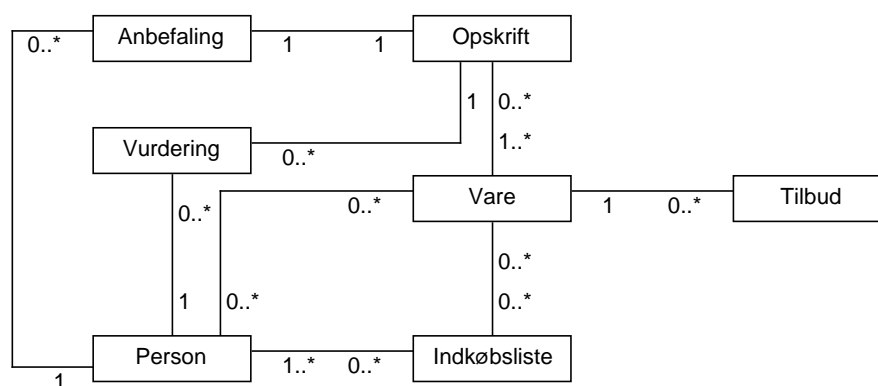
Vi har fundet frem til de forskellige hændelser der sker i problemområdet. Ud fra disse laves en hændelsestabel, der beskriver hvilke klasser forskellige hændelser påvirker. Formålet med at identificere hændelserne samt at analysere disse i en hændelsestabel, er at forstå problemområdet bedre og dermed hjælpe med forståelsen for hvordan en løsning ville kunne designes for at afhjælpe de problemer der findes i problemområdet. Desuden kan tabellen hjælpe med strukturen på klasserne. Hvis 2 klasser har alle de samme hændelser, kan der ofte foretages ændringer og dermed opnå en bedre struktur.

	Tilbud	Indkøbsliste	Opskrift	Vare	Person	Vurderinger
Vare tilføjet til indkøbsliste		+		+	+	
Vare fjernet fra indkøbsliste		+		+	+	
Vare aftjekket på indkøbsliste		+		+	+	
Opskrift valgt ???	+	+		+	+	
Tilbud oprettet	+	+	+	+		
Tilbud aktiveret	+	+	+	+		
Tilbud udgået	+	+	+			
Vare tilføjet til overvågning	+			+	+	
Vare fjernet fra overvågning	+			+	+	
Overvågningsvare på tilbud	+			+	+	
Del indkøbsliste		+			+	
Indkøbsliste oprettet		+			+	
Indkøbsliste slettet		+			+	
Vurdering givet			+		+	+
Anbefaling givet			+		+	+

Tabel 3.1. Hændelsestabel. Viser hvilket klasser, problemområdets hændelser påvirker.

Hændelsestabellen i Tabel 3.1 viser både hvilke hændelser der findes i problemområdet, samt hvilke klasser de påvirker. Hvis tabellen læses vandret kan det ses at klasser der bliver påvirket af mange hændelser er klasser som "Indkøbsliste", "Vare" og "Person".

Ud fra analysen indtil nu kan der dannes et overblik, over klassernes interne interaktion, såvel som hvilke hændelser, involverer hvilke klasser. Denne information kan vi nu bruge til at lave en struktur over klasserne i problemområdet.



Figur 3.1. Klassediagram over problemområdet.

3.3.3 Struktur

Klassediagram som ses på Figur 3.1, dette beskriver forholdet mellem de forskellige klasser som findes i problemområdet. Diagrammets sammenhænge er dannet ud fra hændelsestabellen, og beskrivelserne af klasserne. Følgende afsnit gennemgår diagrammets sammenhænge.

Personer i indkøbssituationer kan lave indkøbslister, disse indkøbslister kan været ejet og administreret af en enten en eller flere personer. En indkøbsliste kan bestå af nul til mange varer. En vare kan være på tilbud i mere end en butik, og derfor have nul til mange tilbud. En vare kan desuden indgå i en opskrift, og er derfor forbundet med nul til mange. Desuden har opskrifterne en til mange varer på listen over ingredienser. Varen kan også være tilføjet til overvågningslisten hos en person, så derfor har personer og varer en nul til mange relation på hinanden. En person i problemområdet kan, vurderer en opskrift, men kan give mange vurderinger, og derfor kan en person give nul til mange vurderinger. En vurdering gives til en opskrift alene, imens en opskrift kan have mange vurderinger, eller ingen vurderinger. Anbefalinger består af en opskrift, imens personen kan have nul til mange anbefalinger.

På baggrund af denne analyse kan systemets implementation designes, men først skal der foretages en analyse af anvendelsesområdet.

3.4 Analyse af anvendelsesområdet

Dette afsnit tager udgangspunkt i metoder fra "Objektorienteret Analyse og Design" og benytter disse til at analysere anvendelsesområdet, dette omfatter brug af systemet, funktioner i systemet samt grænsefladen der er tilknyttet.[3] Afsnittet skal give et overblik over funktionaliteten af systemet, samt formidle hvordan brugeren interagerer med systemet.

3.4.1 Brug

Denne del af analysen har til form at fastlægge interaktion mellem systemet og aktører. Dette gøres ved at identificere brugsmønstre for aktørernes aktioner i systemet.

Aktører

I dette IT-system er der blevet identificeret to aktører. Den første værende datakilden, eTilbudsavisens API, hvor tilbudende hentes fra og den anden værende de brugere, som benytter systemet. For disse to aktører er der således udarbejdet en aktørtabel tabel 3.2 på modstående side, der giver overblik over hvilke brugsmønstre der er, vertikalt opsat, samt hvilke aktører er relevant for disse, opsat horisontalt.

Bruger

Formål: En person, som ønsker at bruge en eller flere af IT-systemets funktionaliteter, til at hjælpe med planlægning af mad og indkøb.

Karakteristik: Systemets brugere har meget varierende erfaring med IT-systemer, samt er bredt ud over mange forskellige aldersgrupper, majoriteten er dog mellem 18 - 30 år og har middel erfaring med IT.

Aktører	Bruger	eTilbudssavis
Login	✓	
Listehåndtering	✓	
Søgning	✓	
Indstil præferencer	✓	
Vurder opskrift	✓	
Se anbefalinger	✓	
Hent tilbud		✓

Tabel 3.2. Aktørtabel. Viser hvilke aktører er involveret i hvilke brugsmønstre

Eksempler: Bruger A er en 47-årig familiefar, der kun har smartphone, da det er arbejdstelefonen på givet af arbejdet. A benytter systemet til at handle ind på vej hjem fra arbejde, hvor indkøbslisten lavet af konen eller datteren bruges som guide i supermarkedet. A vil således gerne kunne tilgå listen fra telefonen, så der ikke er behov for at køre hjem og hente den på papirsformat.

Bruger B er en 21-årig universitetsstuderende, som for nyligt er flyttet hjemmefra. B har meget erfaring med IT, og bruger de dagligt til at navigere rundt på internettet. B har let ved at navigere rundt i systemet og bruge dets funktionaliteter til at lave besparelser på det allerede lave budget, samt at undgå at få pasta til aftensmad hver dag.

eTilbudssavisen

Formål: eTilbudssavisen har til formål at gøre tilbudsdata tilgængelig for systemet, dette sker igennem API. Dette giver information om navnet på tilbuddet, pris, periode og butik. Da dataene der hentes igennem API'en kan være af meget svingende kvalitet, filtreres det, så kun forståeligt tilbudsdata kommer igennem.

Karakteristik: eTilbudssavisen er pålidelig med dataene der sendes, kvaliteten af dataene kan dog svinge meget, og eTilbudssavisen tilbyder ingen fleksibilitet i dets arbejde, hvilket resulterer i noget ubrugligt data som filtreres fra.

Eksempel: eTilbudssavisen gør 2.000 tilbud tilgængelig, heri er nogle på et format der ikke klargøre hvad der er på tilbud, men blot giver en række mærkenavne som er på tilbud. En sådan uforståelig række bliver filtreret ud såvel som andre uforståelige tilbud, og systemet ender tilbage med 1.337 brugbare tilbud, der kan vises til brugeren.

Brugsmønstre

For en yderligere beskrivelse af de funktionaliteter i systemet, som vedrører en given aktør, modelleres en række brugsmønstre, dette er de samme brugsmønstre som ses i aktørtabellen på tabel 3.2. Hver enkelt af disse mønstre vil blive beskrevet igennem en brugsmønstrerespecifikation. Det er ikke alle mønstre som ses på denne liste, nogle af disse er sammentrækninger af flere andre mønstre, som alene virker simple og repetitive at beskrive. Andre er udeladt da de ikke passer helt ind i en sammentrækning, men variationen fra det mønster og andre brugsmønstre ellers beskrevet, er så minimal, at mønstret er anset som værende ubetydeligt at beskrive.

Brugeridentifikation

Brugsmønster:

Objekter:

Funktioner:

Listehåndtering

Brugsmønster:

Objekter:

Funktioner:

Søgning

Brugsmønster:

Objekter:

Funktioner:

Tilpas præferencer

Brugsmønster:

Objekter:

Funktioner:

Vurder opskrift

Brugsmønster:

Objekter:

Funktioner:

Se anbefalinger

Brugsmønster:

Objekter:

Funktioner:

Hent tilbud

Brugsmønster:

Objekter:

Funktioner:

3.5 Kravspecifikation

Formål: Et formelt dokument som fortæller hvilke funktioner som det endelige produkt skal have.

3.5.1 Krav til systemet

1. Systemet skal tilgås ved brug af en internetbrowser, altså skal det være en web-app.
2. Systemet skal benytte aktuelle tilbud fra diverse dagligvarebutikker.
3. Systemet skal have mulighed for at overvåge tilbud på varer valgt af slutbrugeren.
4. Systemet skal have mulighed for at indeholde en personlig indkøbsliste
 - a) Integreret med tilbud.
 - b) Indkøbslisten deles mellem enheder
 - c) Indkøbslisten deles mellem brugere fra samme husstand.
5. Systemet skal kunne præsentere opskrifter for brugere.
 - a) Systemet skal kunne modtage feedback på de opskrifter brugerne giver.
 - b) Systemet skal kunne anbefale opskrifter på baggrund af:
 - i. Madvaner (varieret kost)
 - ii. Bedømmelse på opskrift
6. Systemet skal have en række præferencer pr. bruger.
 - a) Systemet skal kunne fjerne forslag om eksempelvis kød til vegetarianer.

3.5.2 Krav til brugergrænsefladen ("UI")

1. Systemets UI skal være på dansk.
2. Systemet skal kunne anvendes på forskellige enheder
 - a) Computere, $\geq 11"$ og > 768 px i højden og > 1280 px i bredden.
 - b) Tablets, $> 6"$ og $< 11"$
 - c) Smartphones, $\leq 6"$
3. Systemets UI skal være responsivt og tilpasse sig den anvendte platform.
4. Systemets UI skal anvende design principperne "proximity" og "consistency".
5. Systemets UI skal anvende en top-menu til den øverste hierarkiske navigation.
 - a) Top-menuen skal være tilgængelig fra alle sider i applikationen.
6. Systemets UI skal anvende direct mapping, dvs. ikoner som repræsenterer handlinger (metafor).
7. Systemet skal tilbyde en fortryd-knap, i henhold til princippet "Effectiveness".

3.5.3 MoSCoW

MoSCoW-analysen benyttes for at vurdere forskellige funktioner og krav imod hinanden.

De opdeles i fire kategorier: Must have, Should have, Could have og Won't have. Deraf kommer navnet MoSCoW.

Ved hjælp af MoSCoW'en kan vi arrangerer disse krav i kategorier alt efter hvor vigtige de er. Baggrunden for denne ragering er henholdvis baseret på den feedback interviewsne i afsnit 2.3 på side 11 gav og en vurdering fra gruppens side. Vores vurdering er foretaget på baggrund af estimater om hvor tidskrævende hver funktion vil være at implementerer holdt op imod hvor vigtig denne funktion virker for vores respondenter, samt om funktionens relevans i forhold til projektets mål.

Must have

- Indkøbslister
- Tilbudsintegration
- Varer
- Overvågning af tilbud
- Opskrifter
- Desktop interface

Should have

- Deling af indkøbslister
- Notifikationer
- Vurdering af opskrifter
- Anbefalinger
- Mobil interface
- Tilbud -> Varer

Could have

- E-mail notifikationer
- Brugerlavede opskrifter
- Navigation af butik

Won't have

- Kalorietæller
- Stregkode integration
- Overvågning af køleskab

Del B

Udvikling

FiXme Fatal!

1

4.1 MVC-mønsteret

Et af de standardiserede design mønstre, som bruges af mange udviklere er MVC-mønsteret - som står for *Model-View-Controller*. MVC-mønsteret har til formål, at dele systemet op i tre komponenter, nemlig *Model*, *View* og *Controller*. Denne segregering adskiller således "forretnings-logik", "input-logik" og "UI-logik", og gør herved systemet mere fleksibelt, samt fremmer muligheden for at udvikle parallelt på de forskellige komponenter. Dette kan være nyttigt i udviklingen af systemet, men også efter udgivelsen, idet blandt andet "UI-logik" har det med at blive ændret oftere end for eksempel "forretnings-logik". Opdelingen hjælper også til at skabe overblik over koden, og gør det nemmere at udføre tests på systemet. [5]

Nedenfor beskriver vi de tre komponenter.

Model

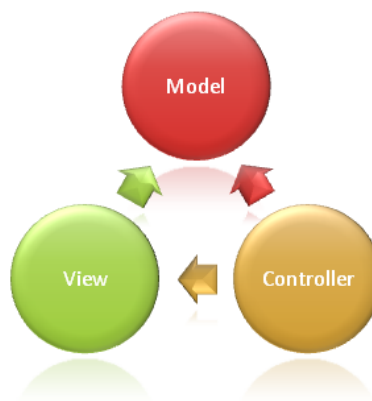
De objekter, der udgør model-laget skal indeholde den før omtalte "forretnings-logik", samt alle data der skal modeleres i systemet. Data'en, i form af objekter, gemmes oftest i en database eller fil, og er helt skjult for brugeren i den forstand, at al repræsentation af modellen foregår igennem view-delen af MVC-mønsteret.

View

Der er igennem forskellige views, at brugeren får præsenteret brugergrænsefladen - også kaldet UI(user-interface). Derfor giver det også mening, at placere "UI-logikken" i denne del af MVC-mønsteret. Typisk bliver et views indhold genereret ud fra data fra en model. Et eksempel på dette ville være visning af en liste af objekter ud fra en model, der indeholder netop en liste.

Controller

Når det kommer til interaktionen mellem brugeren og systemet, er det controlleren der påtager sig opgaven. Derfor er det også i de forskellige controllerere, at vi finder "input-



Figur 4.1. MVC-mønsteret

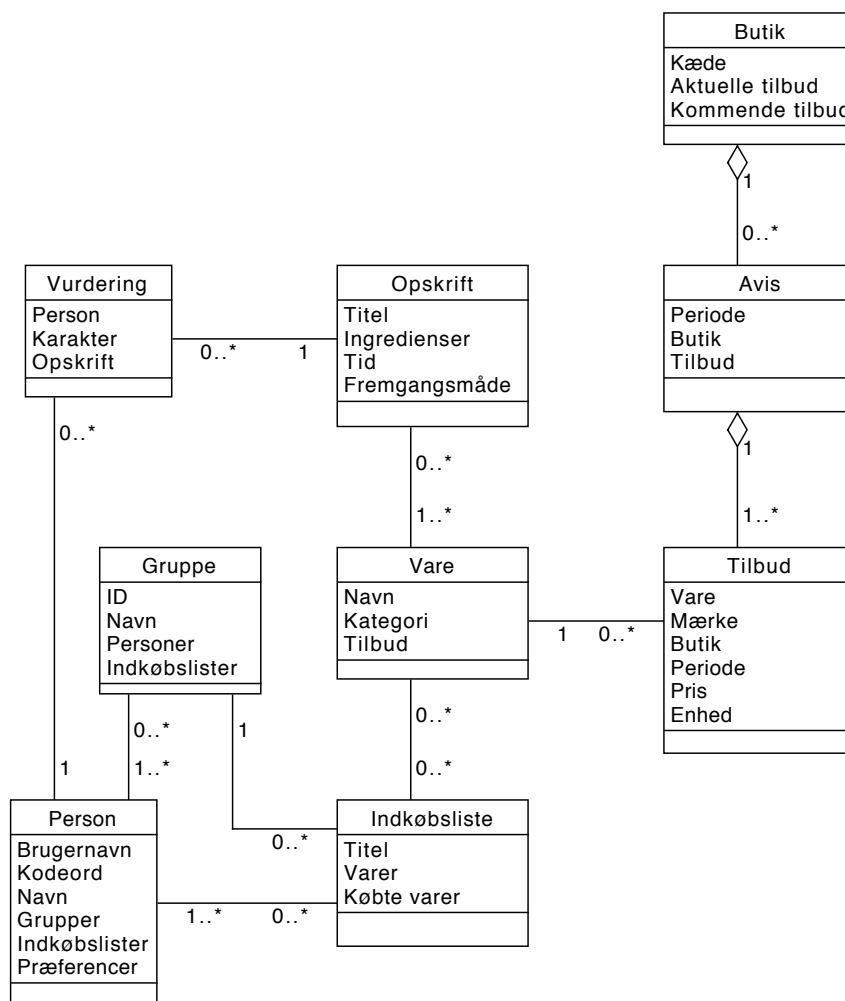
¹FiXme Fatal: indledning her - morten

logikken”. Her bestemmes ud fra input fra brugeren hvilke data der skal arbejdes med i hvilken model og også hvilket view, der skal præsenteret for brugeren. Med dette kan vi også se, at viewet ikke indeholder noget logik og al manipulation af data altså foregår gennem controller komponenten.

4.2 Klassediagram

For at illustrere modellaget i vores MVC-mønster, har vi produceret et klassediagram (se figur 4.2 nedenfor) i UML, der simplificerer strukturen. Det skal bemærkes, at klasserne og felterne er på dansk i diagrammet, og på engelsk i selve koden af programmet.²

FiXme Fatal!



Figur 4.2. UML klassediagram for modellaget i MVC-mønsteret

4.2.1 Klasserne

Vi vil gennemgå klasserne, der optræder i klassediagrammet, og beskrive deres relationer samt de felter de indeholder.

²FiXme Fatal: Der er ingen adgangsting på dette, private, protected, public etc. Er det bevist? Derudover er det ikke klart hvilken datatype hvert felt er, også om det er en liste eller ej. - Troels

Person

Person-klassen i modellen, er den der holder styr på brugeren og dennes basale attributter - herunder brugernavn, kodeord og kaldenavn. Ydermere er det vigtigt, at objektet kan indeholde informationer om personens madpræferencer og vurderinger af opskrifter; dette er med til at give person-klassen en mere intim vinkel, og så at sige bedre afspejle den virkelige person, samt fungere som grundlag for anbefaling af opskrifter. Det er naturligvis også vigtigt for et program, der omhandler bl.a. indkøbslister, at holde styr på en persons indkøbslister og lignende. Til dette har person-klassen to felter der hedder henholdsvis, grupper og indkøbslister. Begge felter er lister³, der holder styr på personens relationer til netop grupper og indkøbslister - En person kan altså have relationer til flere grupper og flere indkøbslister, hvilket også kan ses ud fra de indtegnede relationer i klassediagrammet.

FiXme Fatal!

Gruppe

Klassen "Gruppe", bruges i programmet som en hjælpe-klasse, der binder flere personer sammen om en eller flere indkøbslister. På den måde er det, gennem denne klasse, muligt at dele indkøbslister med andre personer; Man kunne forestille sig en situation i en husstand, hvor flere personer handler ind, og det derfor ville være fordelagtigt, hvis man kunne være fælles om en indkøbsliste. Gruppe-klassen indeholder to attributter der muliggør identifikation: Et ID, der skal være unikt for den enkelte gruppe, så programmet kan skelne mellem grupper; og er navn, der kan sætter af personer i gruppen, så netop personerne kan se forskel på de grupper de er med i. Herudover har klassen to lister, der hver især holder styr på relationerne til henholdsvis personer i gruppen og indkøbslister delt i gruppen. En gruppe har relationer til en eller flere personer, og kan godt eksistere uden indkøbslister.

Indkøbsliste

Indkøbslister repræsenteres i systemet som klassen "Indkøbsliste", og indeholder ID, titel og to lister med varer. ID'et garanterer, at alle indkøbslister er unikke og kan skelnes fra hinanden på systemsiden - akkurat ligesom det gøres i Gruppe-klassen. Også i Indkøbsliste-klassen har vi et titel felt, så brugeren også kan navigere mellem forskellige instanser. De to lister af varer, holder styr på henholdsvis; varer som bliver tilføjet til listen, og varer som har været tilføjet men er blevet markeret som købt. Der er således kun styr på "overstregede" varer - altså dem der er købt, og ikke varer som bliver slettet fra listen. En indkøbsliste kan godt eksisterer uden varer, idet brugeren skal kunne oprette lister, inden der er taget stilling til, hvilke varer der skal bruges. Indkøbsliste har kun relation til en gruppe, men kan godt have relationer til flere personer.

Vare

Vare-klassen indeholder tre felter: Et navn på varen, som er generisk, for eksempel "Letmælk" og "Cola" istedet for "Arla Letmælk" og "Pepsi"; en kategori, der beskriver varen, for eksempel "Mejeri" eller "Pålæg"; og en liste over tilbud, der holder styr på, hvis og hvor varen eventuelt er på tilbud. En vare behøver ikke være på en indkøbsliste eller opskrift for at eksistere⁴, og kan have relationer til nul til mange tilbud.

FiXme Fatal!

³FiXme Fatal: Som nævnt tidligere er dette ikke klart fra diagrammet. - Troels

⁴FiXme Fatal: kan det passe?

Tilbud

Et tilbud som instans af Tilbud-klassen, indeholder felter der beskriver: Varen, mærket, hvilken butik tilbuddet befinder sig i, hvilken periode tilbuddet gælder i, prisen på tilbuddet, og hvilken enhed/mængde tilbudet er i. Ud fra disse attributter er det muligt at identificere tilbudet, og brugeren kan tage stilling til, om det er relevant. Et tilbud relaterer til netop en vare og aggregerer fra Avis-klassen. Denne aggregering betyder, at et tilbud ikke kan eksistere uden en avis.

Avis

Avis-klassen beskriver mere eller mindre blot en række tilbud i en butik. Dette gøres med tre felter, nemlig: Perioden som aviser, og dermed tilbuddene gælder for; butikken som avisen tilhører; og en liste med de tilbud som er i avisen. En avis kan have et til mange tilbud og aggregerer fra en butik. Der skal altså findes en butik, før der kan være en avis.

Butik

En butik beskrives i Butik-klassen med tre attributter. Det er, kæden, aktuelle tilbud og kommende tilbud. Kæden bruges til at identificere butikken - det kunne for eksempel være "Fakta" eller "Netto". De aktuelle og kommende tilbud, repræsenteres af to aviser, og en butik kan derfor have nul til mange relationer til Avis-klassen.

Opskrifter

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Vurdering

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

FiXme Fatal!

4.3 eTilbudsavis API

⁵ For at kunne indhente infomationer om placering af butikker og deres tilbud, anvendes der et API. Dette API udbydes af den danske virksomhed eTilbudsavis.dk, og er af typen REST.

4.3.1 REST

REST står for: *Representational state transfer*. Det er en metode hvorved man opbygger måden som et API. Der findes fire metoder som svarer til de fire som findes i CRUD ("CREATE, READ, UPDATE, DELETE"). Disse metoder er: GET, POST, PUT og DELETE.

GET returnerer data typisk som JSON eller XML og med HTTP svarkoden 200 (OK). eTilbudsavis' API sender JSON. Svarer til READ i CRUD.

POST bruges til at oprette nye ressourcer. Svarer til CREATE i CRUD.

PUT bruges til at opdatere infomationer, eksempelvis til at forlænge en session. Svarer til UPDATE i CRUD.

DELETE bruges til at fjerne ressourcer. Svarer til DELETE i CRUD.

4.3.2 Brug af API

eTilbudsavis' API er kun et halv-offentligt API, dvs. at man skal have en API-nøgle ("APIKEY") og dertilhørende hemmelighed ("Secret"). API'en er placeret på <https://api.etilbudsavis.dk/v2/>. [2]

For at kommunikere med API'en fra C#-kode anvendet værktøjet RestSharp. [6] For at kunne bruge infomationen laves der en klasse, som svarer til det JSON som API'en sender tilbage.

Session

For at kunne bruge API'en skal man først oprette en session. En session består af en API-nøgle, den tilsvarende hemmelighed ("Secret"), en token og en signatur. Har man en API-nøgle kan man kalde API'et sessions og få en token. Den token kan man kombinere med sin hemmelighed og generere SHA-256 hashen af dette som er signaturen.

Eksempelvis vil apikaldet som opretter en session returnere:

```

1 {
2     "token": "00hdcx7fnysn6541",
3     "expires": "2013-03-03T13:37:00+0000",
4     "user": null,
5     "provider": null,
6     "permissions": {
7         "guest": [
8             "api.public",
9             "api.users.create"
10        ]
11    }

```

⁵FiXme Fatal: Dette afsnit er under udvikling ... - Troels

12 }

Listing 4.1. "POST til sessions api'en med APIKEYen"

Dette svar kan oversættes til en klasse i C#.

```
1 public class Session
2 {
3     public string Token { get; set; }
4     public string Expires { get; set; }
5     public string User { get; set; }
6     public string Provider { get; set; }
7     public Permissions Permissions { get; set; } // List<string> Guest
8 }
```

Listing 4.2. "Klassen "Session" som svarer til retur JSONet sendt fra /v2/sessions' APIet"

Hvis dette kald skulle gøres ved brug af C#-kode så vil det kunne gøre på følgende måde, med RestSharp.

```
1 /* [...] */
2 /* Create a RestClient*/
3 var client = new RestClient("https://api.etilbudsavis.dk");
4
5 /* Initiate the first request to get a session */
6 var SessionRequest = new RestRequest("v2/sessions", Method.POST);
7 SessionRequest.AddParameter("api_key", Global.Apikey);
8
9 /* Map the response to the class "Session" */
10 IRestResponse<Session> response2 =
11     client.Execute<Session>(SessionRequest);
12
13 /* Save the response in an object */
14 Session sessionobj = response2.Data;
15
16 /* Generate a signature and set it to the global variable */
17 Global.Signature = EncryptionHelper.SHA256(Global.Secret +
18     Global.SessionToken);
19 /* [...] */
```

Listing 4.3. "C#-kode som opretter en RestClient og anvender den til at oprette et objekt med felter som svarer til JSON dataet givet fra API'en"

Efter eksekveringen af dette kode, vil der være et objekt af datatypen "Session" kaldt sessionobj, som indeholder de samme felter som givet i JSON i listing 4.1 på forrige side; derudover er signaturen ("Global.Signature") også beregnet og gemt.

Fællestræk for den rasterende API

Der findes en række fællestræk for brugen af API'en. Centralt sendes SessionToken'en og signaturen som http-headers. Derudover genanvendes de samme 3 parametre ofte: længdegrad, bredegrad og radius. Disse er et udtryk for en placering af og afstand til butikker.

Store, Catalog og Offer

Det er muligt at spørge API'en om hvilke butikker som er inden for en given radius af en placering som beskrevet ovenfor. Det vil med C#-kode se ud som følger:

```
1  /* [...] */
2  /* Create request for getting stores near a coordinate */
3  var storeRequest = new RestRequest("v2/stores", Method.GET);
4  storeRequest.AddParameter("r_lat", latitude);
5  storeRequest.AddParameter("r_lng", longitude);
6  storeRequest.AddParameter("r_radius", radius);
7  storeRequest.AddHeader("X-Token", Global.SessionToken);
8  storeRequest.AddHeader("X-Signature", Global.Signature);
9
10 var storesResult = client.Execute<List<Store>>(storeRequest).Data;
11 /* [...] */
```

Listing 4.4. "C#-kode som anvender "/v2/stores" delen af API'en til at hente butikker."

storesResult er af datatypen List<Store> hvor "Store" er klassen som svarer til det JSON API kaldet returner. Dette er udformet som følger:

```
1  public class Store
2  {
3      public string id { get; set; }
4      public string ern { get; set; }
5      public string street { get; set; }
6      public string city { get; set; }
7      public string zip_code { get; set; }
8      public Country country { get; set; }
9      public string latitude { get; set; }
10     public string longitude { get; set; }
11     public string dealer_url { get; set; }
12     public string dealer_id { get; set; }
13     public Branding branding { get; set; }
14     public string contact { get; set; }
15     public List<string> category_ids { get; set; }
16 }
```

Listing 4.5. "Klassen Store"

Hvor Branding, Country og Pageflip klasserne er:

```
1  public class Branding
2  {
3      public string name { get; set; }
4      public string url_name { get; set; }
5      public string website { get; set; }
6      public string logo { get; set; }
7      public string logo_background { get; set; }
8      public string color { get; set; }
9      public Pageflip pageflip { get; set; }
10 }
11
12 public class Pageflip
```

```
13 {  
14     public string logo { get; set; }  
15     public string color { get; set; }  
16 }  
17  
18 public class Country  
19 {  
20     public string Id { get; set; }  
21     public string UnsubscribePrintUrl { get; set; }  
22 }
```

Listing 4.6. "Klasserne Branding Pageflip og Country"

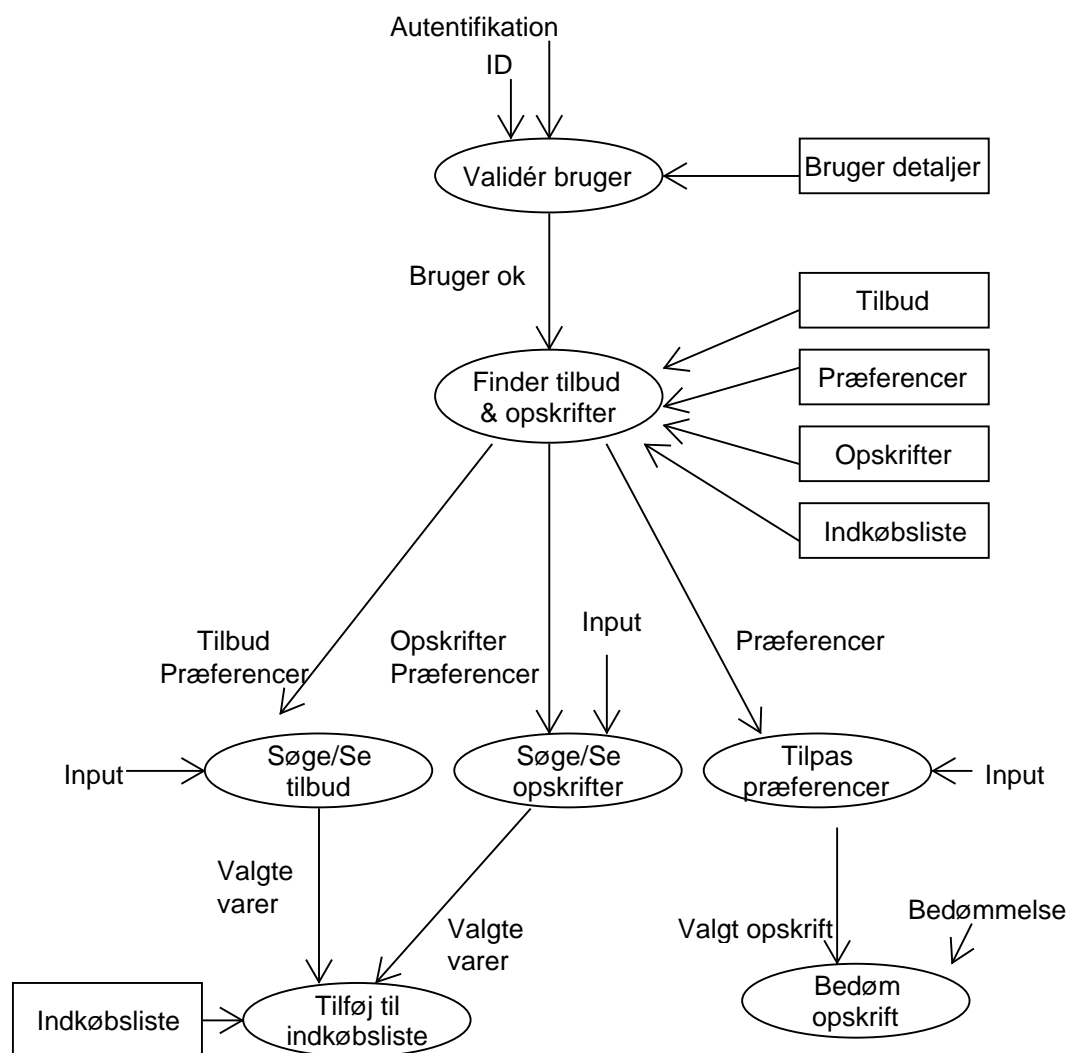
Hele denne process er gentaget og tilpasset Catalog og Offer API'erne for at kunne indhente den information de kan give. Det er hermed muligt at hente informationer om tilbud og butikker.⁶ FiXme Fatal!

⁶FiXme Fatal: Dette er et helvede at beskrive på en måde som giver mening, så alt det ovenstående skal nok skrives om...

Del C

Refleksion

Data-Flow diagram 5



Figur 5.1. UML Data-Flow diagram

¹FiXme Fatal: Hvad er formålet med denne figur? Hvorfor er den med, bidrager den med noget og i så fald hvilket? - Troels

Konceptuelle scenarier 6

1. Tilgå systemet

For at tilgå vores system kræver det en enhed med internetadgang; dette kunne typisk være en eller flere af følgende: computer, tablet, smartphone. For at benytte alle funktioner, heriblandt en indkøbsliste man kan tage med sig, kræver det et mobilt apparat med internetforbindelse, her ville en smartphone være mest oplagt.

2. Finde opskrifter til brug

Brugeren skal kunne benytte en database over opskrifter, der findes i systemet. Hvis brugeren har valgt nogle bestemte præferencer i forhold til opskrifter, dette kunne eksempelvis være fiskefri opskrifter, så vil databasen kun vise de relevante opskrifter. Når brugeren har benyttet en opskrift, eller hvis brugeren kender en opskrift i forvejen, kan brugeren vælge at bedømme denne. Ud fra en række bedømmelser vil systemet finde anbefalinger på opskrifter, som forventes at falde i brugerens smag.

3. Finde tilbud til indkøb

Personer med adgang til systemet vil kunne søge aktuelle tilbud i vores service, disse tilbud vil kunne personliggøres ud fra en persons præferencer. Personen vil også have mulighed for at sætte alarmer eller overvågning op på forskellige varer eller kategorier. Når personen har fundet et tilbud, vil det kunne tilføjes til en elektronisk indkøbsliste.

4. Benytte indkøbsliste

Personer, der benytter systemet, skal kunne tilføje varer til deres indkøbslister, enten via tilbuds-funktionen, tekst-input eller gennem ingredienslisten fra en opskrift. Indkøbslister skal kunne deles i husstanden, så man kan oprette fælles indkøbslister.

5. Sætte præferencer

En bruger af systemet skal kunne sætte præferencer for opskrifter og tilbud, hvis en bruger eksempelvis kun vil se tilbud på økologisk, har allergier eller er vegetar, kan dette sættes under funktionen 'Præferencer'.

Del D

Referencer

Litteratur

- [1] Aarstiderne. Aarstiderne, 2014. URL <http://aarstiderne.dk>. *Side 10*
- [2] eTilbudsavis. etilbudsavis, 2014. URL <http://engineering.etilbudsavis.dk/eta-api/>. *Side 29*
- [3] Lars Mathiassen, Andreas Munk-Madsen, Peter Axel Nielsen, Jan Stage. *Objekt Orienteret Analyse & Design*. Marko, 3. udgave edition, 2001. *Side 13, Side 15, Side 18*
- [4] Mads Nyborg Anneberg. Lette fødevareindkøb vinder frem, 2013. URL <http://www.business.dk/detailhandel/lette-foedevareindkoeb-vinder-frem>. *Side 9*
- [5] Microsoft ASP.NET Team. Asp.net mvc overview, 2009. URL <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>. *Side 25*
- [6] RestSharp. Restsharp, 2014. URL <https://github.com/restsharp/RestSharp>. *Side 29*
- [7] SuperBest. Online supermarked, 2014. URL <http://www.superbest.dk/online-supermarked>. *Side 10*

Figurer

2.1	Tilbudssaviser på eTilbudssavis.dk	7
2.2	MitFakta	8
2.3	Forside fra Prototypen, der blev brugt i forbindelse med vores 2. runde af interviews	11
3.1	Klassediagram over problemområdet.	17
4.1	MVC-mønsteret	25
4.2	UML klassediagram for modellen i MVC-mønsteret	26
5.1	UML Data-Flow diagram	35

Tabeller

2.1	Nogle butikskæder med smartphone apps samt deres funktionaliteter.	8
3.1	Hændelsestabel. Viser hvilke klasser, problemområdets hændelser påvirker.	17
3.2	Aktørtabel. Viser hvilke aktører er involveret i hvilke brugsmønstre	19