

BACHELOROPPGAVE 2008

Business og IT studiet

EK227E 003

Veien fra ide til lønnsom internettbasert applikasjon

Rapport tilrettelagt og skrevet av:

Frank Tjernmoen

Geir O Zakariassen

Trond Torkildsen



Veien fra ide til lønnsom internetbasert applikasjon

I. Forord

Forfatterne

Vi er tre studenter på Bachelorstudiet: Business og IT ved Handelshøgskolen i Bodø som har samarbeidet om vår avsluttende Bachelor oppgave. Bakgrunnen for at akkurat vi tre samarbeider rundt dette prosjektet bunner i en felles ide om å utvikle en egen web applikasjon spunnet rundt et internetbasert forretningskonsept.

Rapportkonvensjon

Før vi presenterer ide og konsept vil vi informere om en konvensjon for denne rapporten. Vedlagt denne rapporten, ref vedlegg 3, følger en ordliste som gir en kort forklaring til termer som vi tror ikke er godt kjent for alle. Ordene og uttrykkene som er tatt inn i ordlisten er merket på følgende måte den første gangen de er omtalt i selve rapporten: Eksempel .

Bacheloroppgaven

Oppgaven er oppsummert i overskriften ” *Veien fra ide til lønnsom internetbasert applikasjon*”. Utleddning av problemstillinger er gjort mot et konkret webutviklingsprosjekt, heretter kalt Forbrukertipset.no¹. Problemstillingene er videre brutt opp i tre fokusområder:

- Forretningsområdet
- Brukerområdet
- Teknologiområdet

En tilnærming til denne tredelingen, samt listing av problemstillinger ses under innledningen.

Samarbeidspartnere

Vi har vært våre egne oppdragsgivere og således håndtert dette internt. Vi vil dog takke Høgskolen i Bodø og da spesielt vår veileder, Henrik Dvergsdal, for den støtte vi har fått der. Videre vil vi takke våre informanter som velvillig og på kort varsel stilte opp til brukertest.

Bodø den 22. mai 2008

Frank Tjernmoen

Geir O Zakariassen

Trond Torkildsen

¹ Utviklingsprosjektets url: <http://www.forbrukertipset.no/>



II. Sammendrag

Oppgavens hensikt

Bacheloroppgaven har til hensikt å belyse det vi mener er sentrale problemstillinger knyttet til å etablere en egenutviklet web applikasjon ut fra forretningsmessige og brukermessige behov og strategier. Dette gjør at vi kan gjøre bruk av et bredt spekter av de kunnskaper som studiet har gitt oss, samt etablere ny og viktig kunnskap innen enkelte felt.

Omfang av undersøkelser og metodebruk

Det har vært nødvendig å tilegne seg kunnskaper som ligger utenfor studiets pensum både når det gjelder utviklingen av forretningsområdet og teknologiområdet. Utviklingen av forretningsområdet har støttet seg på aktuelle problemområder som det finnes oppdatert informasjon om på Internett. Her kan søkemotoroptimalisering nevnes som et godt eksempel. Dette er et svært viktig område for oss sett både i et forretningsmessig og teknologisk perspektiv og hvor oppdatert og relevant informasjon må hentes fra internettkilder og ikke to år gamle bøker.

Videre hadde vi et behov for å gjøre brukerundersøkelser for å belyse om vår applikasjon, i form av en prototyp, tilfredstilte brukerens behov for informasjon. Det ble etablert en egnet testmetodikk som det er redegjort for i kapittel 2, metode.

Utviklingen av selve applikasjonen har vært av en eksperimentell karakter. Vi som utviklere har måttet tilegne oss kompetanse underveis og realisert denne kompetansen i en helhetlig applikasjon, prototypen. Vi har således fulgt en egendefinert tilnærming til metodikk som det er redegjort for under kapitel 2, metode.

Resultater

Ut fra definerte problemstillinger har vi slått fast at vår ide virker å være teknisk løsbar med den kompetansen vi hadde og i tillegg har klart å tilegne oss underveis i prosjektet. Den forretningsmessige visjonen har blitt bekreftet å være gyldig ut fra resultat av brukertester og videre undersøkelser gjort under forretningsområdet. Brukertester og beste praksis sett fra et brukerperspektiv har også gitt klare utviklingsretninger for en endelig applikasjon.

Planer videre

Bacheloroppgaven avslutter første fase i etableringen av det nye Forbrukertipset.no. Vi har nå en prototyp som må utvikles videre og modnes ut fra føringer som kommer fra forretningsområdet og brukerområdet.



Det må videre jobbes aktivt for å kunne tilby flere produkter fra vår applikasjon, samt tilby enda bedre sorteringsmuligheter for brukeren. En tradisjonell søkemotor basert på fritekst må implementeres slik at denne kan gi gode svarresultater for brukeren. Da vil vi sikre at brukeren også får det som vanligvis er forventet med denne type tjenester. I tillegg har vi konkrete planer om å lansere flere tjenester på nettstedet noe som igjen krever utvikling av nye applikasjoner og en helhetlig integrering av alle applikasjonene på ett nettsted.

Abstrakt

This Bachelor gives a real life approach on how to realize a business vision on the internet through building a web application. This involves a development in three focus areas which are business, usability and technology. We will introduce at prototype of a web application witch demonstrate your capability to solve the business vision technically. Through usability tests we will also try do demonstrate that the vision is understandable for the user.

By exploring the business area we will find our competitive advantages, define our business model and the success criteria's needed to evaluate if this is a money making application. We will use user tests as a method to evaluate our prototype and analyze the results to see if it fulfills our vision and find trouble areas we have to focus on in further development. And finally we will take you through the process of how we made the prototype.



III. Innholdsfortegnelse

VEIEN FRA IDE TIL LØNNSOM INTERNETTBASERT APPLIKASJON	2
1. INNLEDNING	8
1.1. Bakgrunn for valg av tema	8
1.2. Presentasjon av strategitriangel.....	8
1.2.1. Teoretisk forankring av Strategitriangel	9
1.3. Presentasjon av problemstillinger	11
1.4. Oppgavens utforming	12
1.5. Avgrensninger av prosjektet	13
1.5.1. Avgrensninger av Prototyp	13
2. METODE	14
2.1. Forskningsdesign.....	14
2.1.1. Design for enkel brukertesting.....	14
2.1.1.1. Tilnærming til design for brukertesting	14
2.1.1.2. Design for enkel brukertesting av prototyp	15
2.1.1.3. Hvordan skal testen fysisk gjennomføres	17
2.1.1.4. Oppsummering rundt testdesign.....	18
2.1.2. Eksperimentell applikasjonsutvikling.....	18
2.1.2.1. Tilnærming til design for prototyp utvikling	18
2.1.2.2. Mål med-/ og organisering av utviklingsløpet.....	19
2.1.2.3. Oppsummering rundt utviklingsløp.....	20
3. FORRETNINGSOMRÅDET	21
3.1. Konkretisering av problemstilling	21
3.2. Hvilke fortrinn og markedsstrategier kan bidra til lønnsomhet?.....	22
3.3. Hvilke fortrinn må vi fokusere på i forhold til våre konkurrenter?	23
3.3.1. Forretningsside'	23
3.3.2. Forbrukertipsets konsept.....	23
3.3.3. Visjon	23
3.3.4. Prosjektets strategiske målsettinger.....	23
3.3.4.1. Strategiske målsettinger.....	23
3.3.4.2. Administrative målsettinger	24
3.3.4.3. Operative målsettinger.....	24
3.4. Forretningsmodeller.....	24
3.4.1. Eksisterende forretningsmodeller på Internett	25
3.4.2. Vår forretningsmodell.....	27
3.5. Hvilke kostnadseffektive markedsføringstiltak vil passe vår situasjon?	28
3.5.1. Grov markedskartlegging	29
3.5.2. Bransjetall og trender.....	29
3.5.3. Søkemotoroptimalisering, et viktig markedsstrategisk tiltak	32
3.5.3.1. Organiske treff.....	32
3.5.3.2. Betalt trafikk.....	33
3.5.3.3. Hvordan optimalisere betalt trafikk for best mulig return of investment. [ROI]	34
3.5.4. Hvilken strategi kan vi så treffe vårt marked best med.....	36
3.6. Hvordan definere lønnsomhet - Innsats, tid og fortjeneste?	38
3.6.1. Verdisette innsats.....	39
3.6.2. Break Even	41
3.7. Etikk og internett	43
3.7.1 Innledning.....	43
3.7.2 Bruk av informasjonskapsler	43
3.7.2.1 Etisk gråsoner i forretningsmodell.....	44
3.7.3 Vår holdning til denne gråsonen.....	44
3.8. Oppsummering	45



4.	BRUKEROMRÅDET	46
4.1.	Presisering av problemstilling	46
4.2.	Tilnærming – Løs historikk rundt brukerperspektivs utvikling	46
4.2.1.	Våre inngangsparametere med tanke på brukeren i prototyp utvikling	48
4.3.	Teste om visjon og konsept er løst i prototypen	49
4.3.1.	Målsetninger, hypotese og begrensinger for brukertest av prototyp	49
4.3.2.	Redegjørelse for gjennomførte brukertester av prototyp	50
4.3.3.	Konkrete ufordringer som funnene i brukertestene oss	52
4.4.	Identifisering av brukerkonvensjoner og om disse følges i prototypen	52
4.4.1.	Identifisering av grunnprinsipper	52
4.4.2.	Konkrete ufordringer opp mot brukerkonvensjoner	54
4.5.	Oppsummering	55
5.	TEKNOLOGI OMRÅDET	57
5.1	Presisering av hovedproblemstilling	57
5.2	Hvordan løse andre strategiske momenter fra forretningsområdet?	57
5.2.1	Søkemotoroptimalisering (SEO)	58
5.2.1.1	Teknisk kvalitet	58
5.2.1.2	Andre momenter	59
5.3	Hvordan løse identifiserte krav fra Brukerområdet?	60
5.3.1	Grensesnitt og konsekvenser	60
5.3.2	Kvalitet og tilgjengelighet	60
5.4	Realisering av visjonen (prototyp utvikling)	61
5.4.1	Tjenersiden – Utviklingsløpet	61
5.4.1.1	Plan for- og utvikling av serverfunksjonalitet	62
5.4.1.1.1	Databaseløsning	62
5.4.1.1.3	Hvordan lese og sette sammen xml dokumenter	63
5.4.1.1.4	Hvordan fremskaffe egen tilleggsdata	64
5.4.1.1.5	Legge ønskede data inn i databasetabellene	65
5.4.1.2	Behandle forespørsler fra klienten	69
5.4.1.2.1	Hvilke input må serveren ha for å respondere korrekt	71
5.4.2	Klientsiden – Utviklingsløpet	74
5.4.2.1	Grensesnittet	74
5.4.2.2	Dynamisk lasting av data	76
5.4.2.3	DOM Manipulasjon	79
5.4.2.4	Oppsummering	81
6.	KONKLUSJONER VEDRØRENDE PROBLEMSTILLINGENE	83
6.1.	Konklusjoner vedrørende forretningsområdet	83
6.1.1	Forretningsområdets overordnede problemstilling	83
6.1.2	Forretningsområdets første underproblemstilling	83
6.1.3	Forretningsområdets andre underproblemstilling	83
6.1.4	Forretningsområdets tredje underproblemstilling	84
6.2.	Konklusjoner vedrørende brukerområdet	84
6.2.1	Brukerområdets overordnede problemstilling	84
6.2.2	Brukerområdets første underproblemstilling	85
6.2.3	Brukerområdets andre underproblemstilling	85
6.2.4	Brukerområdet oppsummert	86
6.3.	Konklusjoner vedrørende teknologiområdet	86
6.3.1	Teknologiområdets overordnede problemstilling	86
6.3.2	Teknologiområdets første underproblemstilling	87
6.3.3	Teknologiområdets andre underproblemstilling	87
6.3.4	Teknologiområdet oppsummert	87
7.	REFERANSER LITTERATUR/RESSURSER	88
	Trykte kilder	88
	Internett kilder	88
	Annet	89
8.	APPENDIKS - RAPPORTENS VEDLEGG	90



IV. Figurliste

Figur 1 - The Information Systems Strategy Triangle	10
Figur 2 - Forbrukertipset's strategitriangel	11
Figur 3 - Konseptuell fremstilling av prosjektets hovedprosesser	21
Figur 4 - Konseptuell illustrasjon av affiliasmodellen	27
Figur 5 - Forbrukertipset.no's valgte forretningsmodell	28
Figur 6 - Grafisk fremstilling over utvikling i netthandel.	30
Figur 7 - Norske netthandlers valg for å finne kilden til varen de vil kjøpe	31
Figur 8 - Illustrasjon av substandig innovasjon og disruptiv innovasjon	37
Figur 9 - Åtte sentrale parameter i forbindelse med disruptive innovasjoner	38
Figur 10 - Grafisk fremstilling av nullpunktomsetning i kr og antall.	42
Figur 11 - Startsidene slik den fremsto ved brukertesting	48
Figur 12 - Navigasjonskonvensjoner illustrert	54
Figur 13 - Det nye produktsorteringsapplikasjonen	55
Figur 14 - Det nye produktsorteringsapplikasjonen, første progresjon.	56
Figur 15 - Det nye produktsorteringsapplikasjonen, andre progresjon.	56
Figur 16 - Yahoo's graderingstabell for nettlelere.	59
Figur 17 - Prinsippskisse for samspelet mellom klient og tjener.	61
Figur 18 - ER diagram som viser identifiserte behov for databasetabeller, samt deres knytninger.	63
Figur 19 - Prinsipp modell for DOM prosessering av xml.	63
Figur 20 - Interesseområdene til første kolonne i vår applikasjon.	64
Figur 21 - Dataflyt modell for skript: fillMainSubCatDb.php	66
Figur 22 - ER diagram for databasetabell "cat_prod_td"	66
Figur 23 - Dataflyt modell for skript: dataToLinkTab.php	67
Figur 24 - Dataflyt modell for skript: addCatToLinkTab.php	68
Figur 25 - Dataflyt modell for skript: readToProdTab.php	69
Figur 26 - ER diagram for databasetabell "pr_price_cat"	70
Figur 27 - Grovskisse av prosessen med å motta og besvare forespørsler fra klient.	71
Figur 28 - Kolonnefordeling i prototypen	74
Figur 29 - Flytdiagram over prosessen med å fylle kolonnene med data.	75
Figur 30 - (X)HTML strukturen for første kolonne	76
Figur 31 - Styling til kolonnestrukturen	76
Figur 32 - Ajax versus tradisjonell klient/tjener spørringer	78
Figur 33 - XHR objektets rolle	78
Figur 34 - Kode for XHR objektet	79
Figur 35 - Presentasjon av første kolonne.	80
Figur 36 - Neste trinn i navigasjonsprosessen.	80
Figur 37 - Kildekoden første kolonne med DOM generering.	81

V. Tabelloversikt

Tabell 1 - Konvertering og Return of investment	36
Tabell 2 - Totalt investerte midler fram til lanseringstidspunkt 1 jan 2009	40
Tabell 3 - Budsjett fra lansering 1. januar 2009, basert på konverteringsrate på 1,5 %	41
Tabell 4 - Budsjett fra lansering 1. januar 2009, basert på konverteringsrate på 2,0 %	42
Tabell 5 - Økonomiske opplysninger for nullpunktberging	45

VI. Nøkkelord

Affiliate, Brukertest, Brukervennlighet, Prototype, Strategiområder, Webapplikasjon, Webutvikling.



1. Innledning

1.1. Bakgrunn for valg av tema

Bachelorgruppens medlemmer var allerede i gang med et internettbasert prosjekt i begynnelsen av 2008. Dette arbeidet var i all hovedsak diskusjonsbasert rundt ideer om å utvikle en nettjeneste som tilbyr produkter fra flere leverandører via *affiliate-network*, heretter kalt partnernetverk. Forretningen skulle kun fokusere på verdensveven og inntjening skje i form av provisjon på salg generert fra vår side. Det ble så klart for oss at de problemstillingene vi diskuterte, samt konseptet i seg selv, egnet seg godt for en bred tilnærming til en Bacheloroppgave innen Business og IT. Samtidig kunne en gjennomarbeiding som en Bacheloroppgave bidra til å avgjøre om dette var forretningsideer som det var verdt å jobbe videre med.

Etter en grundig gjennomgåelse av vårt internettprosjekt, heretter kalt Forbrukertipset.no og sett i lys av en tenkt studieoppgave, kom vi frem til et opplegg som ble godkjent av Høgskolen i Bodø som vår Bacheloroppgave.

1.2. Presentasjon av strategitriangel

Vi har et utviklingsprosjekt i gang som både er forretningsutvikling og utvikling av en nettapplikasjon samtidig. Etter at prosjektet også ble en Bacheloroppgave ville det også innebære utvikling av en studierapport i tillegg til de oppgavene som ligger under de nevnte områder.

Styrken ville for oss være at vi kunne bruke arbeidet med Bacheloroppgaven som en grundig undersøkelse rundt de forhold som vi mener er viktig for å lykkes. Det ble drøftet ulike strategier for hvordan vi best mulig kunne få Forbrukertipset.no satt ut i livet med høyest mulige odds for suksess. Vi ble enige om at innsatsfaktorene var den tiden vi var villig til å investere i prosjektet og den kunnskapen vi hadde på starttidspunktet, pluss den kompetansen vi ville tilegne oss i prosjektperioden. Prosjektperioden er på kort sikt satt til å være lik Bacheloroppgavens innleveringstidspunkt som er den 22. mai 2008, og på litt lengre sikt en evaluering av prosjektet sommeren 2009. Slike forretningsmessige rammer vi nevner her er også eksempler på strategier som må besluttes i et rent forretningsmessig perspektiv og vi kommer tilbake til det i avsnitt 1.2.1.



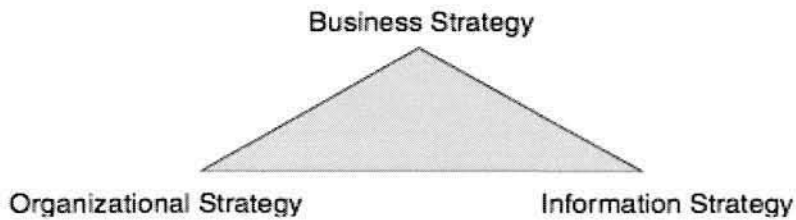
De første samtaler var av mer teknologisk og teknisk karakter. Vi hadde kreative samtaler om hvordan navigasjonsprosesser på internett kunne gjøres bedre. Vi snakket om hvordan vi kunne bruke Ajax på et smartere vis enn det vi hadde sett til nå.

Sett i forhold til forbrukere og handling av varer på nettet diskuterte gruppen også ulike tekniske løsninger for å tilby dette utover dagens *affiliate* modeller med XML feed. Parallelt med disse samtalerne drøftet vi også hva som ville kreves fra et rent brukermessig perspektiv. Hva karakteriserer brukeren? Hvordan teste om våre ideer faktisk ble forstått og helst foretrukket av brukeren? Disse spørsmålene er etter vår mening strategiske spørsmål som drives frem av de forretningsmessige strategiene, men som bør ses på for seg selv og atskilt fra de rent tekniske og teknologiske valgene som Ajax og XML feeds tilhører. Samtidig er de knyttet til hverandre så vi vil se på de som to separate strategier, men som samtidig er to områder som hører til en overordnet forretningsstrategi.

Dersom de tre områdene vi har snakket om her behandles som egne strategier hvor forretningsstrategien driver brukerstrategien og teknologistrategien fremover vil vi kunne ha bedre fokus på oppgavene i utviklingsløpet. De er avhengig av hverandre og må være i en balanse, men det gir et bedre utgangspunkt for konstruktive tanker om hva som vil kreves for å konkurrere i markedet. Det gir oss også et skarpere fokus på de oppgaver som må gjøres for å nevne noen av de fordelene dette gir oss. Idemessig er dette en tankemodell som vi selv ”oppdaget” i de innledende samtalerundene, men vi ser også at det faktisk eksisterer en slik triangelmodell fra pensum vi har hatt under Business og IT studiene ved Handelshøgskolen i Bodø. Den passet ikke rett inn i vår kontekst så vi har tilpasset den til vår kontekst og vi vil belyse vår overførte betydning av den under neste avsnitt som tar hånd om den teoretiske forankringen.

1.2.1. Teoretisk forankring av Strategitriangel

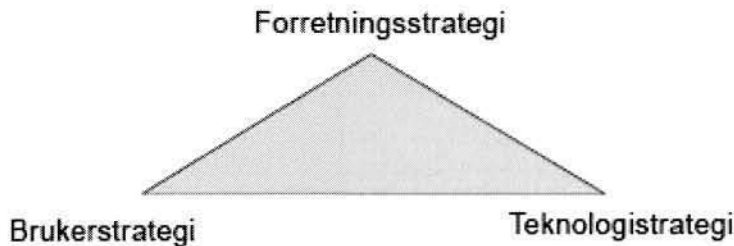
(Pearlson, Saunders, 2006) introduserer en strategitriangel i boken *Managing & Using Information Systems*. Denne består av forretningsstrategi, organisasjonsstrategi og informasjonsstrategi. Figuren under viser modellen slik den presenteres i nevnte bok.



Figur 1 - The Information Systems Strategy Triangle

Denne triangelen er et enkelt rammeverk for å øke forståelsen av i hvilken grad Informasjonssystemer påvirker organisasjoner. Den kalles "Information Systems Strategy Triangle" (ibid: 20) fordi den relaterer og knytter forretningsstrategiene med organisasjonsstrategiene og informasjonsstrategiene sammen. Modellen foreslår tre nøkkelpunkter rundt strategier. Det er fordi suksessfulle virksomheter har en overordnet forretningsstrategi som driver både organisatoriske strategier og informasjonssystemstrategier. En virksomhets beslutninger om nye ansettelser og andre organisatoriske strategier så vel som beslutninger rundt hardware, applikasjoner og andre informasjonssystemstrategier drives alle av virksomhetens overordnede forretningsstrategi. Det viktige her er at dersom en av strategiene endrer seg så må de andre strategiene endre seg deretter. Hvis for eksempel forretningsstrategien sier at vi skal bruke teknologi som konkurransemessig fortrinn vil informasjonsstrategien måtte innovere og stadig finne nye løsninger for å opprettholde posisjonen. Videre vil endringer i informasjonsstrategien innebære konsekvenser for de to andre områdene. De tre strategiene må derfor hele tiden være i en balanse. Satt inn i vår sammenheng vil vi umiddelbart se nytten av denne filosofien. På samme måte som forretningsstrategi, organisasjonsstrategi og informasjonsstrategi representerer nøkkelområder i en virksomhet av en viss størrelse, så representer vår forretningsstrategi, brukerstrategi og teknologistrategi nøkkelområder i vårt prosjekt. Forbrukertipset.no er ikke en organisasjon, ikke i dag. Det er ikke det samme som å si at det ikke eksisterer organisatoriske beslutninger, men de kan utmerket tas hånd om av forretningspunktet. For oss vil dermed organisasjonsstrategi byttes ut med brukerstrategi. Videre vil informasjonsstrategi endres til teknologistrategi. Så vi nedskaleres og setter dette rammeverket inn i vår sammenheng, men vi ivaretar fortsatt kraften til verktøyet fordi vi også vil trenge å opprettholde en balanse mellom utviklingen av disse områdene.

Vi vil også måtte fokusere sterkt på disse områdene for å bli konkurransedyktige, og vi har også tre komponenter som krever vedlikehold og samhandling. Vår modifiserte strategitriangel kan ses i figuren under.



Figur 2 - Forbrukertipset's strategitriangel

1.3. Presentasjon av problemstillinger

Gruppen ble enige om at vi måtte identifisere problemstillinger som både var interessant sett fra et teoretisk og akademisk perspektiv, men som også var interessant sett fra et utviklingsmessig perspektiv. Vi ville forene teori med praksis på en slik måte at begge behovene kunne gi næring til hverandre slik at grunnlaget for en optimalisering var tilstede. Som uteksaminerende Business og IT studenter følte vi også at det var viktig å sette fokus på problemstillinger som tar for seg både organisatoriske, økonomiske og informasjonsteknologiske spørsmål. Etter disse hensyn har vi etter mange diskusjoner kommet frem til en overordnet spørsmålsstilling.

Hva kreves for å realisere en lønnsom nettbasert applikasjon som Forbrukertipset.no?

Denne spørsmålsstillingen er utgangspunktet for våre problemstillinger. Videre er det slik at problemstillingene er fordelt på strategitriangelens tre områder og har alle resultert i konkrete underproblemstillinger. Vi kommer tilbake til hvordan disse problemstillingene skal søkes løst og på hvilken måte de er knyttet til de respektive strategiområdene både i avsnitt 1.4 (Oppgavens utforming) og kapittel 2 (Metode), i tillegg vil det være en presisering av problemstillingene under hvert hovedkapittel.



Følgende problemstillinger er identifisert for denne Bacheloroppgaven:

Hvilke fortrinn og markedsstrategier kan bidra til lønnsomhet?

- Hvilke fortrinn må vi fokusere på i forhold til konkurrentene?
- Hvilke kostnadseffektive markedsføringstiltak vil passe vår situasjon?
- Hvordan definere lønnsomhet: Innsats, tid og fortjeneste?

Oppfyller prototypen forretningsvisjonen sett fra et brukerperspektiv?

- Skjønner brukerne vårt innovative produktfiltrerings konsept?
- Følger utviklet nettapplikasjon etablerte brukerkonvensjoner?

Hvilke teknologiske og tekniske forutsetninger ligger til grunn for en realisering av applikasjonen?

- Hvordan realisere visjonen? – Teknisk fungerende prototyp.
- Hvordan løse andre strategiske momenter fra forretningsområdet?
- Hvordan løse identifiserte krav fra brukerområdet?

Vi er av den oppfatning at dette er problemstillinger som er svært omfattende og vi har ikke ambisjoner om en rettfærdig dybdegrad og fordeling mellom disse. Detaljnivå vil derfor variere med hva gruppen anser som å være tilstrekkelig for å oppnå den overordnede spørsmålsformuleringen. Det er også slik at ressurs og tidsbegrensninger ikke gir rom for optimal detaljgrad.

1.4. Oppgavens utforming

Vi vil bruke strategitriangelen som et rammeverk for hele utviklingsprosjektet, fra visjon til ferdig utviklet applikasjon og etablert forretning. Samtidig vil hovedproblemstillinger og underproblemstillinger ha sitt utspring i de tre strategiområdene, og triangelen er derfor også å anse som overordnet ramme for strukturering av besvarelsen til Høgskolen i Bodø. Oppgaven utformes slik at hvert område har sin hovedproblemstilling med tilhørende underproblemstillinger. Hvert område starter med en presisering og redegjørelse for denne problemstillingen og kapittelet avslutter med en oppsummering av de viktigste funnene. Etter at de tre områdene er gjennomgått avslutter vi med et eget kapittel for konklusjoner. Vi starter med forretningsområdet fordi det er her de overordnede målsettinger og strategier ligger, deretter følger vi opp med brukerområdet, og til slutt behandler vi teknologiområdet.



Det er naturlig å ha denne rekkefølgen på kapitlene og vi anbefaler å lese rapporten i denne rekkefølgen, dog er det ikke et krav da føringer fra et område til et annet vil bli forklart. Før vi starter med de tre strategiområdene gjennomgår vi avgrensninger som er gjort i prosjektet og deretter et kapittel om metodebruk.

1.5. Avgrensninger av prosjektet

Et slikt prosjekt som dette favner veldig bredt. Vi mener problemstillingene som er nevnt maler et tydelig bilde av de aspekter vi ønsker å prioritere, men det er også en rekke forhold som gruppen anser som både relevante og viktige, men som det enten ikke er tid eller ressurser til å omhandle i denne rapporten eller som kan virke noe begrenset. Det er for eksempel et viktig satsningsområde å prioritere søkemotor optimalisering, men dette er ikke en oppgave om søkemotoroptimalisering. Det er også mange flere faktorer inne bildet som avgjør graden av suksess, og selv om vi hadde hatt mulighet til å undersøke alle så er det ingen garantier for at det vil lykkes. Dog mener vi at det bør belyses i mer detalj hva vi avgrenser, men dette er mer gjeldene for prototypen og vi behandler det i neste avsnitt.

1.5.1. Avgrensninger av Prototyp

Prototypen er en teknisk løsning basert på en ide om enklere sortering/søk i store mengder data over internett. For å begrense omfanget av prototypen ble det tidlig avgjort at prototypen kun skulle håndtere produkter tilgjengelig fra Tradedoubler i eget xml format. Siden Tradedoublers xml struktur danner utgangspunkt for modellering og skripting er deres xml struktur presentert i vedlegg 6. Det ble også besluttet å ikke implementere fritekstsøk på klientsiden selv om en tilfredsstillende funksjonalitet er på plass på tjenersiden.



2. Metode

Metode kan defineres som læren om å samle inn, organisere, bearbeide, analysere og tolke informasjon på en systematisk måte. Metode velges ut fra problemstillingen som skal belyses og hensikten med oppgaven. Det skilles mellom to hovedformer; kvantitativ og kvalitativ metode. ”Dataen er kvantitativ dersom den er målbar eller kvalitativ når den sier noe om ikke-tallfestbare egenskaper” (Ghauri og Grønhaug, 2002: 85).

2.1. Forskningsdesign

”Forskningsdesign er alt som knytter seg til en undersøkelse” (Johannessen et al. 2004: 73). For å kunne besvare våre problemstillinger har vi benyttet oss av forskjellige metodikkdesign som det vil bli redegjort for senere.

I forbindelse med utredning rundt alle problemstillingene har det vært behov for sette teoretiske referanserammer, det være seg gjennom bruk av studiets pensumlitteratur eller annen litteratur som vi har funnet relevant og formålstjenelig.

Dette representerer sekundærdata, ”Sekundærdata er informasjon innsamlet av andre, av årsaker som kan være andre en ens egne” (Ghauri og Grønhaug, 2002: 76).

Fordelen ved å bruke sekundærdata er at det er kostnadseffektivt og tidsbesparende, siden dataene allerede foreligger. Ulempen er at det kan være vanskelig å avgjøre om dataene er korrekte. Vi behandler ikke litteraturstudie noe mer inngående i denne rapporten da vi ikke anser det for å tilføre leseren noen ytterligere verdi.

2.1.1. Design for enkel brukertesting

2.1.1.1. Tilnærming til design for brukertesting

Brukertester av den art vi legger opp til tilhører kvalitativ metode og design. I et forskningsperspektiv vil vår gjennomføring av brukertesting ligge nærmest opp til fenomenologi.

”Som kvalitativt design betyr en fenomenologisk tilnærming å utforske og beskrive mennesker og deres erfaringer med og forståelse av, et fenomen” (Johannessen et al. 2004: 80). Fenomenet her er vår prototypapplikasjon som vi ønsker individuelle tilbakemeldinger på hvordan oppfattes i testsettingen. Vi hadde også behov for et mer konkret design for gjennomføring av brukertester opp mot applikasjonen.



Krug (2006:s.133) gir følgende redegjørelse for hva en brukertest av en web applikasjon er: ”I en brukertest skal en bruker om gangen presenteres for en web side, eller en prototyp av en side eller eventuelt bare noen skisser av en web side ide. Vedkommende skal (a) finne ut hva dette er eller (b) gjøre spesifikke oppgaver”.

Da det vanligvis stilles store formelle krav til testing i et vitenskapelig perspektiv har brukertesting av websider/applikasjoner ofte blitt nedprioritert. Testing etter mer vitenskaplig metodikk krevde mye tid og planlegging, mange informanter, eksperter til både å utføre testene samt å tolke resultatene.

Dette bidro til store kostnader for applikasjonseier både økonomisk og tidsmessig. Dette førte til at brukertesting i liten grad ble gjennomført ved utviklingsprosjekter under en hvis størrelse.

I 1989 tok Jakob Nielsen til ordet for en enklere test metodikk, Nielsen(1989), denne ble dokumentert å være effektivt, men fremdeles relativt omfattende og kostbart for små prosjekter som måtte realiseres fort. Krug (2006) tok det derfor et skritt videre og introduserte en svært enkel, men etter hans mening effektiv brukertesting. Denne tilnærmingen har vært utgangspunktet for vår brukertesting opp mot prototypen.

2.1.1.2. Design for enkel brukertesting av prototyp

For å belyse de viktige forskjellene mellom tradisjonell brukertesting og den enkle brukertesten lansert av Krug har vi oversatt og tatt med et tabellarisk oppsett som viser dette (Krug, 2006). I tillegg gir tabellen en god oversikt over hva dette designet innebærer.

	Tradisjonell testing	Enkel & kostnadseffektiv testing
Antall informanter pr brukertest.	Vanligvis åtte eller flere informanter.	Tre eller fire informanter.
Hvordan rekruttere informanter.	Nøye utvelgelse ut fra en definert målgruppe.	Hvem som helst som har kjennskap til internett.
Miljø for gjennomføring av test.	Bruker lab med et tilknyttet observasjons rom med enveis speil.	Hvilket som helst plass hvor en kan teste uforstyrret.
Hvem bør utføre testene.	En erfaren profesjonell innenfor webbruker teori.	Enhver fornuftig og tålmodig person som kjenner applikasjonen.



Når må planlegging starte.	Test må planlegges flere uker i forveien for å sikre tilgang til lab, og de riktige informantene.	Nærmest ingen planlegging nødvendig.
Forberedelser før test.	Utarbeide en skisse for testing. Denne må diskuteres og tilpasses. Endelig oppsett protokollføres.	Finn ut hva du skal vise og hva du eventuelt skal spørre om.
Hva og når tester du.	Hvis ikke budsjettet er uvanlig stort gjøres normalt bare en større test like før lansering.	Mange små tester gjennom utviklingsperioden.
Kostnader	25.000 til 75.000 kroner. I tillegg kommer mye tidsbruk.	0 – 2.000 kroner (maks kr 500 til hver informant).
Hva skjer etterpå	Eiere mottar en 20 siders rapport en uke etter testene og må ut fra den avgjøre hva som skal endres.	Interessenter kan analysere og bli enige om endringer over lunsj samme dag.

Essensen av dette blir at ved små utviklingsprosjekt, som vårt, bør det testes ofte og enkelt.

Testene kan med fordel utføres av de samme som tar hånd om selve utviklingen.

Informantene kan være nærmest hvem som helst som ikke er bundet til prosjektet i utgangspunktet. Dette knytter brukerens behov og meninger tett opp til utviklingsarbeidet og endringsbehov kan raskt fanges opp og innordnes i videre utvikling. En oppnår dermed en vinn, vinn situasjon med tanke på både de fremtidige brukerne, samt eierne av applikasjonen. Dessuten sparer en både tid og kapital ved en slik gjennomføring, noe som er av stor betydning for vårt prosjekt.

Kritikk av metoden

Den største kritikken mot en slik fremgangsmåte for testing har gjerne som utgangspunkt at det er for få og for tilfeldig valgte informanter. Denne kritikken kan gjøres gyldig i to tilfeller:

1. Hvis det skal gjøres bare en test bør en bruke flere informanter. Åtte informanter har vist seg å gi en god identifisering av problemområder. Flere informanter tilfører sjeldent identifisering av flere problemområder, bare bekreftelser på de funn som allerede er gjort.
2. Hvis applikasjonen er tiltenkt en meget spesifikk gruppe brukere bør en, hvis dette er relativt enkelt, rekruttere informanter som er representative for den gruppen brukere.



2.1.1.3. Hvordan skal testen fysisk gjennomføres

Testen krever lite forberedelser, men det må være avklart hva målet med testen skal være. Det må videre være gjort en avtale med informant om tid og sted. Videre må det tekniske grunnlaget være på plass for gjennomføring og logging av test.

Datainnsamling

Selve testen gjennomføres ved at testpersonen får utlevert et kort notat om selve testen, notat er gjengitt i vedlegg 5. Dette leser informanten gjennom for så å sette seg ved internettmaskinen hvor vår applikasjon er oppe og går med startsidene, se figur 11 under kapitel 4. Informanten skal så, uten hjelp fra testveileder, ta stilling til hva siden kan brukes til og utforske denne fritt. Men det må påses at informanten prøver å komme seg til kolonne fem på minst et produkt. Kolonne fem er der hvor et spesifikt produkt presenteres og hvor det kan gjøres valg om bestilling som tar brukeren til et annet nettsted.

Det er viktig at testpersonen tenker høyt under denne utforskningsprosessen og at alle betraktninger blir registrert.

Etter at informanten har testet siden må han/hun besvare noen enkle kontrollspørsmål, hvis da disse ikke er blitt besvart i utforskningsprosessen. Spørsmålene er som følger.

- La du merke til søkeboksen i bunnen av applikasjonen?
- Var det enkelt å se navigeringsstolpe som ble lagt til i venstre del av skjermbilde etter at du hadde gjort valg i kolonnen ”produkter”, kolonne fire?
- Hva er ditt generelle inntrykk av denne tjenesten?

De dataene vi sitter igjen med etter brukertestene er å anse som primærdata. ”Primærdata er originaldata innsamlet av en selv i forbindelse med egen undersøkelse” (Ghauri og Grønhaug, 2002: 81).

Analyse og rapportering

Utviklingen av teknologi som kan støtte gjennomføring av brukertester har bidratt til at dette kan gjøres av hvem som helst og dokumenteres på en enkel og effektiv måte. Dette gir mulighet for at flere kan vurdere testene i ettertid i en så tett opp til virkeligheten situasjon som mulig. Vi sikter spesielt til programvare som kan registrere informantens bevegelser og valg gjennom å gjøre digitale opptak av skjermbevegelser. I tillegg vil en fra samme programvare få registrert hva informanten og testveileder sier under gjennomføringen.



Dette gir en unik mulighet til å analysere og dokumentere testene raskt og enkelt i ettertid. Vi har valgt å bruke en programvare kalt Camtasia Studio² for å dokumentere de enkelte testene i tillegg til notater som gjøres av testveileder.

2.1.1.4. Oppsummering rundt testdesign

Denne enkle testmetodikken er valgt fordi den forventer å gi oss tilstrekkelige svar for videre utvikling uten at vi må bruke unødig mye ressurser på dette.

Det legges opp til bare en testrunde mot prototypen for å avdekke mangler ved denne.

Videre legger valg av denne enkle testmetodikken føringer om at flere lignende tester bør gjennomføres i det videre utviklingsarbeidet både for å re-teste at våre endringer fungerer, samt avdekke nye problemstillinger som igjen må inn i utviklingsarbeidet.

2.1.2. Eksperimentell applikasjonsutvikling

2.1.2.1. Tilnærming til design for prototyp utvikling

Dette er ikke snakk om eksperiment etter vitenskapelig designmetodikk, men et eksperiment på om vi er i stand til å produsere en applikasjon som går klart utover den kompetansen som Business og IT kurset har tilført oss av applikasjonsutviklingskunnskaper.

Det eksisterer mange modeller for programvareutvikling som ivaretar et metodisk og kvalitetsmessig utviklingsløp og som således kan fungere som designmetodikk.

Disse er i stor grad kommet til for å kunne gjennomføre store komplekse utviklingsoppgaver som involverer mange faggrupper med erfaring og erfaringsdata fra andre prosjekter.

Vi har ikke hatt behov eller tid og erfaring nok til å tilnære oss og ta i bruk slik omfattende metodikk. For å plassere vår arbeidsmåte vil vi si at det har vært en mellomting mellom Code and Fix modellen og eXtreme Programming(XP) modellen.

Code and Fix modellen

Code and Fix modellen stammer fra applikasjonsutviklingens spede barndom og fungerer godt til å løse enkle problemer med enkel kode. Modellen har to faser (Berg Hansen og Hjertø, 2003) som følger:

1. Det programmeres en systembit.
2. Eksisterende kode rettes for feil eller endres med tanke på funksjonalitet eller ny funksjonalitet legges til.

Her ligger altså ingen krav til dokumentasjon eller direkte interaksjon med bruker.

² TechSmith's Camtasia Studio a Full-Motion Screen Recording Software



eXtreme Programming modellen

”Hovedideen bak XP modellen er bruk av beste praksis” (Berg Hansen og Hjertø, 2003: 312). Dette er en modell tilpasset utvikling av mindre applikasjoner mot for eksempel internett. Den baserer seg på å dele utviklingen opp i små deler som kan testes slik at endringsbehov kan innarbeides så raskt som mulig.

Som nevnt tidligere ligger vår tilnærming et sted mellom disse modellene. Det hadde vært ønskelig å tilnærme seg XP metodikk da denne utgjør et mye mer fullverdig design, men det betinger at en har innarbeidede erfaringer og etablert en beste praksis.

Dog vil vi fremheve at vi har trukket brukeren tett inn i utviklingen gjennom testing og vi har også dokumentert utviklingsprosessen relativt grundig. Vi har også gjort tilnærminger til beste praksis gjennom teori for koden både på server og klientsiden.

2.1.2.2. Mål med-/ og organisering av utviklingsløpet

Mål med utviklingsløpet

Målet med prototyp utviklingen har vært å finne løsninger som vi som utviklere kunne håndtere for å realisere ideen for applikasjonen. Dette har betinget at vi som utviklere har måttet tilegne oss nye kunnskaper om hvordan enkelte utfordringer kunne løses og da helst innenfor en teoretisk etablert beste praksis.

Organisering av utviklingsløpet

Utviklingen av prototypen slik den fremstår har vært klart todelt. Klient og tjenerside har blitt utviklet parallelt, men med bruk av forskjellige teknikker og programmerings skript/språk. I tillegg har hovedansvaret for utviklingen av de områdene vært fordelt på forskjellige utviklere. Et tett samarbeid på utviklingssiden så langt har derfor vært på de områdene hvor klient og tjenersiden må utveksle informasjon.

Årsaken til at denne organiseringsløsningen er valgt er fordi vi følte behov for at den enkelte utvikler kunne fokusere på ett spesifikt område slik at dette ble mestret så godt som mulig. Dersom gruppen skulle vært delaktig i alle programmeringsspråk og teknikker vil det alltid være en fare for at vi ikke oppnår nødvendig spisskompetanse.

På den annen side har vi ikke kunnet utnytte de fordeler som for eksempel XP fremhever med parprogrammering. Veldig kort innebærer det at det alltid skal være to programmerere som skriver kode. De skal sitte ved samme maskin og mens den ene skriver så fungerer den andre som diskusjonspartner og kodeinspektør. Fordelene skal være at det skrives færre feil og sikrer at de langsiktige kostnadene blir mindre.



På den annen side så er parprogrammering et av de omstridte elementene i XP (Berg Hansen og Hjertø, 2003: 312). Så det kan tenkes at vi ikke ville fått mer ut av en slik løsning uansett.

2.1.2.3. Oppsummering rundt utviklingsløp

Selv om vi har delt oss inn i tjener og klientsiden med tanke på utviklingsløpet har vi et kollektivt eierskap til hele koden. Vi har lagt opp til tett muntlig kommunikasjon med hverandre undervegs i løpet slik at erfaringsoverføring og kollektivt eierskap ivaretas. Videre har vi strukturert og kommentert koden slik at vi internt kan orientere oss i hverandres kode.



3. Forretningsområdet

3.1. Konkretisering av problemstilling

Følgende hovedproblemstilling og underproblemstillinger er sentrale innen forretningsområdet og hva som må løses for å oppfylle disse.

- Hvilke fortrinn og markedsstrategier kan bidra til lønnsomhet?
 - Hvilke fortrinn må vi fokusere på i forhold til konkurrentene?
 - Hvilke kostnadseffektive markedsføringstiltak vil passe vår situasjon?
 - Hvordan definere lønnsomhet: Innsats, tid og fortjeneste?

Et helhetlig syn på prosjektet med fokus på samhandling mellom de 3 kjerneområdene omtalt i punkt 1.2 vil bidra med å øke muligheten til lyktes med etableringen av vår bedrift.

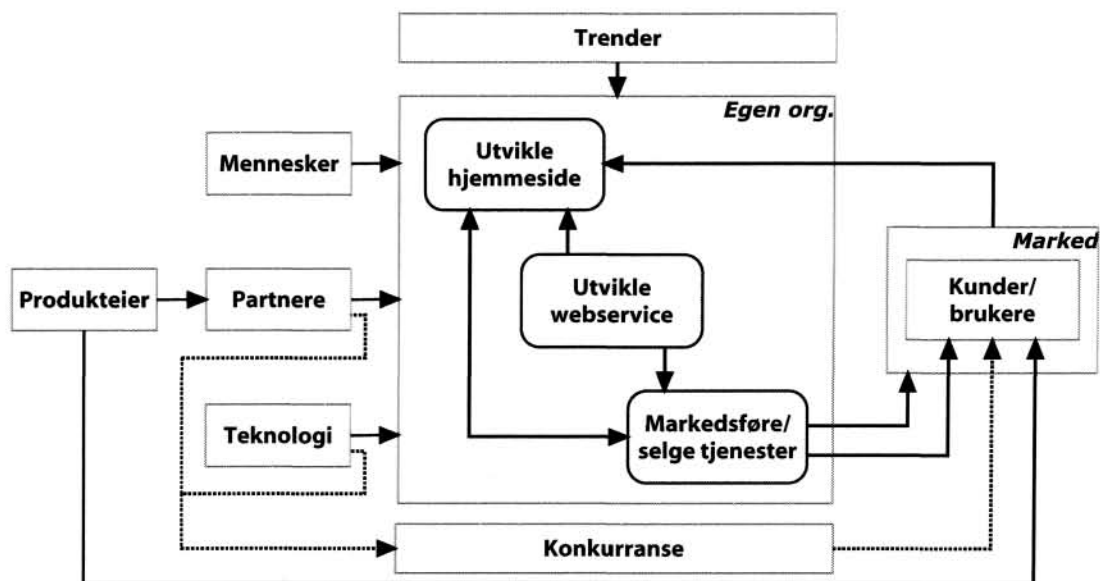
Vår ide om å lansere en brukervennlig nettløsning for produktsøk rettet mot privatmarkedet stiller store krav til at valgte problemstillinger gir nødvendig input til de 2 andre kjerneområdene i strategitriangellet.

Vårt valg om å benytte strategitriangellet som mal for arbeidsmetodikk i prosjektet gir forretningsområdet mulighet til å holde rett fokus med henblikk på de strategiske føreringer som må også svares ut av teknologi og brukerområdet..

Overordnet vil prosjektet omtales som en innovasjon, nytenkende ide' hvor krav til godt

entreprenørskap er avgjørende hvorvidt prosjektet kan realiseres eller ikke.

Prosjektets hovedutfordringer kan i stor grad leses direkte ut fra figur 3 av den konseptuelle fremstillingen av hovedprosessene i prosjektløpet.



Figur 3 - Konseptuel fremstilling av prosjektets hovedprosesser



På toppen står trender, det er da en abstrakt betegnelse på hva som er med å påvirke hva vi produserer, i vårt prosjekt et nettsted som tilbyr en prissammenligningstjeneste fra utvalgte produktområder. Nederst står konkurranse. En stiple linje føres fra partnere og teknologi inn i denne boksen og til kunden, dette illustrer at vi ikke er alene om å bruke partnernettverk som Tradedoubler og lignende tilbydere. Vi kan med dette si at vi konkurrerer om den samme kunden sammen med flere tilbydere.

På høyre side i figur 3 står de støttefunksjonene som hovedprosessene er anhengig av:

Mennesker = Prosjektgruppens medlemmer.

Partnere = Leverandører av produktfeeds³.

Teknologi = Hardware, software samt utvikling av applikasjoner for håndtering av XML.

Produkter = De som leverer produktene, enten det er produsent eller forhandler.

Modellen, figur 3, illustrerer viktigheten av samhandling i prosjektet. Valgte problemstillinger som besvares i dette kapitlet danner nødvendig input i forbindelse med behandlingen av de andre områdene i vårt strategitriangel. Vi ønsker å belyse det som omhandler foretningsområdet i vårt prosjekt med henblikk på tilpassing til internettmarkedet. Prosjektets forretningside, overordnet omtalt i innledningen gir input i forbindelse med videreutviklingen av konseptet og tilhørende strategier.

I vår belysning av det forretningsmessige holder vi hovedfokus på de forhold som har spesiell betydning i sammenheng med en internettsatsning. Forhold som er tilstedeværende i alle forretningsmessige sammenhenger generelt belyses ikke spesielt. Naturlig avgrensning vil være at fullstendig utarbeidelse av forretningsplan, finansieringsplan og markedsplan ikke tas med i vår oppgave. Dette betyr ikke at vi undervurderer viktigheten av disse elementene.

Forretningsområdet gir input til våre andre fokusområder.

3.2. Hvilke fortrinn og markedsstrategier kan bidra til lønnsomhet?

Beskrivelse av forretningside, strategiske valg med tilhørende målsettinger er sentrale forhold som må belyses. Ofte skiller en mellom begrepene visjon, mål og forretningside, disse begrepene henger nøye sammen.

Selv om en visjon kan være noe vag er det viktig å ha en visjon og ha et bevisst forhold til det. Uten noe å strekke seg etter er det vanskelig å sette konkrete strategiske mål.

³ Produktfeeds: Filer i xmlformat innholdene strukturert produktdata



3.3. Hvilke fortrinn må vi fokusere på i forhold til våre konkurrenter?

Våre fortrinn skal være enkelhet, dynamiske løsninger, hurtighet. Kort oppsummert et nettsted som "alle" kan forstå og benytte. For å kunne ta riktige valg er en naturlig prosess å tilegne seg kunnskap innen sentrale områder forbundet med forretningsdrift på internett.

3.3.1. Forretningsside'

Kjernen i forretningssideen er en egenutviklet applikasjon som på en innovativ og dynamisk måte skal hjelpe brukerne å gjøre prissammenligninger i et uoversiktlig marked.

Vår styrke skal ligge i søk og navigasjon, tjenesten skal ha en lav brukerterskel slik at vi kan møte den stadig økende andelen av nye, uerfarne og eldre netthandlere.

3.3.2. Forbrukertipsets konsept

Vårt nettsted skal uten å stille store krav til brukeren gi en svært enkel og intuitiv tilgang til et stort antall produkter og tjenester.

3.3.3. Visjon.

Forbrukertipset.no skal gjennom utvikling av innovative løsninger lage brukervennlige og intuitive webløsninger for sortering og søk i store mengder data. Forbrukertipset.no tar sikte på å bli det foretrukne valg for prissammenligninger i privatmarkedet.

Det skal også søkes utviklingsmuligheter rundt konseptet mot andre markeder.

3.3.4. Prosjektets strategiske målsettinger.

For å kunne følge en målbevisst progresjon i forretningsutviklingen er tidsbestemte målsettinger nødvendig.

Det vil være naturlig å sette overordnede strategiske mål som støtter opp under valgt visjon.

Videre settes administrative og operative tidsbestemte mål og delmål.

Dette vil hjelpe oss å holde fokus i prosjektløpet. Målsettinger sorteres under strategiske, administrative og operative målsettinger.

3.3.4.1. Strategiske målsettinger

Strategiske mål fastsettes på et overordnet nivå, målene kjennetegnes med at de ikke er så konkrete og at de er langsiktige. Overordnet beskriver det som er drivende for de andre typene målsettinger. For oss er et strategisk mål beskrevet slik:

Bli konkurransedyktige i et tøft marked dominert av store aktører.



3.3.4.2. Administrative målsettinger.

Administrative og operative mål er konkrete og av kvantitativ eller kvalitativ art. Disse målene er derved de som er styrende for enkeltaktivitetene i prosjektet. For vårt prosjekt beskriver vi dette slik.

Prosjektet skal generere overskudd innen avslutning av regnskapsåret 2009.

Investerte midler i prosjektet skal generere positive kontantstrømmer fra 2009 og ha en tilbakebetalingstid på maksimalt 3 år.

3.3.4.3. Operative målsettinger.

Prototyp for applikasjon ferdig innen 24.5.2008

Merk. Rapporten detaljbehandler ikke operative målsettinger utover prototyp men prosjektets videre målsettinger nevnes allikevel i forretningsmessig stand.

Applikasjon versjon 01 med tilhørende brukergrensesnitt klar for lansering 1.9.2008

Applikasjon versjon 02 med søk innen reise, hotell og leiebil innen 31.12.2008

Utvikle applikasjon med "prisagent" som gir feedback via e-post og sms til kunden når ønsket pris på et valgt produkt, ferdig innen 1.7.2009

3.4. Forretningsmodeller

For å tilskaffe oss kunnskap om hvilken foretningsmodell(er) vi skal benytte i vår internettsatsning har det stor nytteverdi for oss å tilegne oss kunnskap om overordnede markedsmodeller og kunnskap internettbaserte forretningsmodeller spesielt. Dette vil hjelpe oss å forstå forretningsmessige forhold på internett, samt at kunnskap på dette området vil bidra til økt kreativitet hva angår valg og eventuell videreutvikling av forretningsmodell tilpasset vår virksomhet.

Valg av forretningsmodell vil danne fundamentet som teknologiområdet må bygges rundt. Bruk av internett og internetteknologi er i rask utvikling, litt historikk gir et overblikk på utviklingen. Perarlson og Saunders (2006) viser til data fra forskning i USA hvor en kan se at det tok 7 år før 30 % av befolkningen hadde tilgang til internett, mens det tok henholdsvis 17 og 45 år når det gjaldt tilgang på TV og strøm. Dette viser til høy utviklingstakt allerede i startfasen. I dag har nesten alle i den Norske befolkningen tilgang til internett. Tidligere var det kun store kapitalsterke bedrifter som benyttet elektronisk data interchange (EDI) i forbindelse med forretningstransaksjoner via nettet. Ny teknologi åpner for nye måter å gjøre forretninger på. Generelt om forretningsmodeller kan man si at disse sorteres ut fra hvilke markeder som skal betjenes.



(ibid) skisserer følgende overordnede markedsmodeller:

- B2B - Business to Business, forretningsmodell som driver handel mellom forretningsenheter.
- B2C - Business to Consumer, forretningsmodell som driver handel mellom forretning og forbruker.
- B2E - Business to Employee, forretningsmodell som driver tjenestelevering rettet mot ansatte i forretningsenheter. Eksempelvis pensjonsfond forsikringsordninger.
- B2G - Business to Government, modell som betjener handel spesielt rettet mot myndighet.
- C2C - Consumer to Consumer, modell som betjener handel mellom forbrukere. Eksempelvis Finn⁴ med rubrikkannonser, Qlx⁵ med auksjoner.
- Hybrid - Kombinerer B2B og B2C og / eller C2C. Ofte vil en se at flere modeller benyttes. Dette kan være for å betjene forskjellige kundegrupper. Et nettsted kan for eksempel ha ulike modeller for privatkunder og bedriftskunder.

3.4.1. Eksisterende forretningsmodeller på Internett

De overordnede markedsmodellene gir oss retningen inn mot videre valg av forretningsmodell for vår internettbaserte virksomhet.

Rappa(2008) har identifisert ni hovedkategorier for internettbaserte forretningsmodeller. En kort presentasjon av de ni hovedkategoriene presenteres her, mens en mer utfyllende omtale av disse med tilhørende underkategorier kan finnes i vedlegg 4.

Affiliate:

Modell for salgsmarkedsføring og affiliateprogrammer gjør at e-handelssteder, portaler og informasjonsnettsteder kan oppnå økt salg og trafikk. Ved å bare betale for genererte salg får nettstedene som benytter affiliatepartnere kostnadseffektive salgskanaler

Forretningsmodellen baseres på å tilby provisjon basert på omsetning til partnernettstedene. Bindeledd mellom leverandør og partnernettside er partnernettverk som Tradedoubler, Zanox, Commission Junction og Travelfusion.

Advertising:

Modellen for nettbasert annonsering fungerer som en slags mediekanal. En annonsemodell vil kreve høy trafikk eller høy grad av målretting om den skal være lønnsom.

⁴ <http://www.finn.no> nettsted som startet med C2C modell rubrikkannonser for kjøp og salg

⁵ <http://www qlx.no> nettsted som startet med C2C hvor privatkunder aksjonerer ut produkter

*Brokerage:*

Kobler kjøpere og selgere og gjennomfører transaksjoner. En megler tjener penger på en avgift per transaksjon.

Community:

Modellen betinger kundelojalitet. Brukerne bruker mye tid her og har et emosjonelt forhold til nettstedet. Modellen kan være abonnementsbasert for spesielle tjenester.

Infomediary:

Data om brukere og deres handlemønster har stor verdi. Denne modellen gir kunden noe gratis mot at brukeren legger igjen data. Eksempel kan være å svare på spørsmål i en konkurranse. Spørsmålene formes slik at de ved å analyseres gir nyttig informasjon om brukers handlevaner.

Merchant:

Svært vanlig i B2B markedet. I markedsplassmodellen belastes vanligvis selgeren en transaksjonsavgift basert på den pris som er oppnådd i salget.

Metamediary:

Fungerer som et mellomledd mellom kunder. Konseptet er at dette mellomleddet kobler kjøpere med selgere som så kan utveksle informasjon. Dette er som et virtuelt varehus, men har i tillegg ansvar for gjennomføring av transaksjonen, sporer/følger ordrer og tilbyr fakturerings- og betalingstjenester.

Subscription:

Innholdsbaserte nettsteder med innhold som er gratis å benytte, men som krever registrering for bruk. Ved at brukeren registrerer seg kan en variant være at brukeren godtar at det sendes e-post (reklame).

Slik kan nettstedet bygge opp en verdifull portefølje.

Utility:

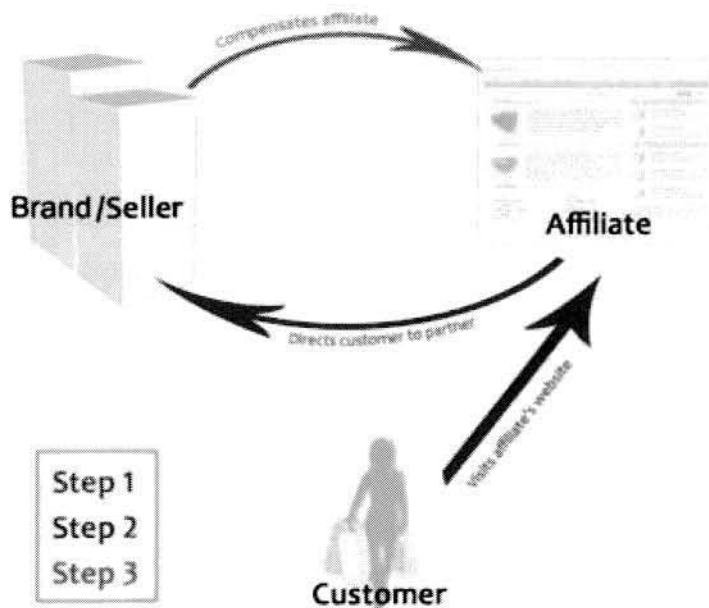
Modellen tar betalt etter forbruk. En typisk utilitymodell kan være internett service provider (ISP) som tilbyr server plass. Bruker betaler etter behov for kapasitet.

Å gjøre bruk av forretningsmodeller er meget nyttig. Det vil være lettere å se hvor en kan sette inn tiltak til forbedringer. Ofte vil det være riktig å benytte kombinasjoner av flere modeller for mest optimal interaksjonen mellom firma og forbruker.

For oss vil en god oversikt over mulige forretningsmodeller bidra til kreativitet i arbeidet med å velge optimal løsning.

3.4.2 Vår forretningsmodell

Affiliate-modellen som benyttes til salgsmarkedsføring gjennom såkalte affiliate-program vil være vår modell i denne fase. Prinsippene en affiliate-modell er bygget rundt kan illustreres som i figur 4 (wikipedia 2008).



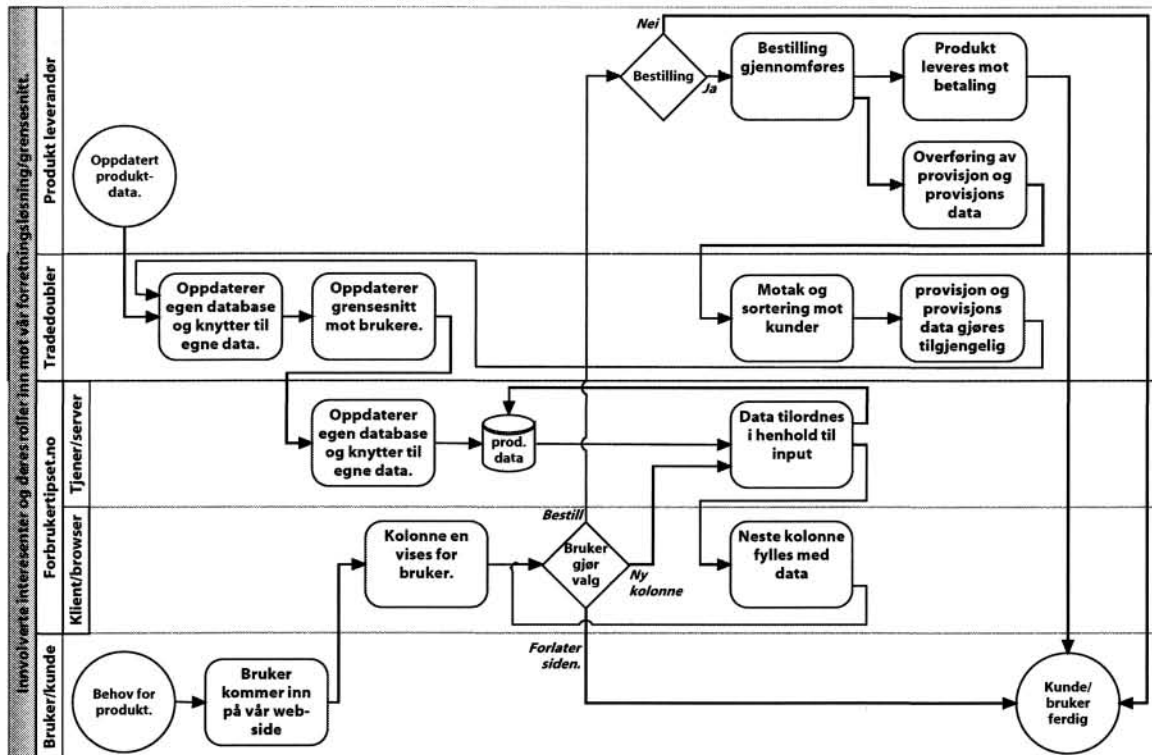
Figur 4 – Konseptuel illustrasjon av affilietmodellen

De partnernettverkene vi er medlemmer av ivaretar det tekniske bindeleddet mellom oss og leverandørene. Typiske forhold som ivaretaes av de som driver partnernettverkene er leveranse av produktdata, annonseformater, statistikk og provisjonsavregning.

De fleste som jobber opp mot disse partnernettverkene benytter ferdige annonseformater, slik som bannere, produktbilder, tekstlinker og autogeneratede produktpresentasjoner og produktfeeds i XML format. Tilgangen til data basert på sistnevnte format, XML, gir oss mulighet til etablere en netttjenste basert på disse avtalene, noe som gir oss tilgang til flere hundretusen produkter og reiser. Med unntak av store aktører er det ikke mange i Norge som tilbyr tekniske løsninger lignende vår applikasjon. Spesielt er det få aktører i forretningsområdet B2C, slik sett har vi en fortrinn her.

Visuell fremstilling av foretningsmodellen er vist i figur 5. Modellen beskriver hvordan bruker, forbrukertipset, tradedoubler og produktleverandør(er) samhandler. Prosesskartet beskriver gangen fra kunde kommer inn på websiden, gjør et valg (søk) på klient siden.

Videre hvordan data lastes fra database på serversiden, eventuell bestilling fortaes. Bestilling med påfølgende provisjonsavregning til oss.



Figur 5 – Forbrukertipset.no's valgte forretningsmodell

Forhold som kan påvirke vår forretningsmodell er å sjekke ut om det er mulig å inngå direkte kontrakter med leverandører. Hva vil dette i så tilfelle kreve av investeringer, utvikling av grensesnitt eller anskaffelse av software som kan håndtere infrastrukturen mellom leverandør og vår applikasjon, samt teknisk løsning for provisjonsberegning.

Videre vurdere utvidelse av modellen, tilby ekstratjenester som nyhetsbrev. Innhold kan være tilbud på varer og tjenester satt etter parameter kunden selv har valgt. Dette vil bidra til flere kanaler (nye verdikjeder) inn et potensielt salg. Sosiale nettverk, for eksempel en blogg(er) innen felt kundene kan ha utbytte av. Eksempel kan være reisetips, erfaringsutvekslinger etc. Også en slik utvidelse av modellen vil bidra til flere kanaler inn mot potensielle salg.

3.5. Hvilke kostnadseffektive markedsføringstiltak vil passe vår situasjon?

Vårt prosjekt skal i utgangspunktet være selvfinansierende. I dette legger vi at vi ikke tar høyde for å gå ut på markedet for å innhente kapital i form av samarbeide med andre inntresenter. Nødvendig egenkapital vil eventuelt bli av personlige midler.



Nettopp av denne grunn er det av stor viktighet for oss å velge de markedsføringstiltak som er optimale og mest kostnadseffektive for oss. Optimalisering rettet mot digital synlighet og optimalisering av annonsekampanjer generert av betalt trafikk vil ha høy prioritet. Dette omhandles grundigere under punkt 3.5.4.

3.5.1. Grov markedskartlegging

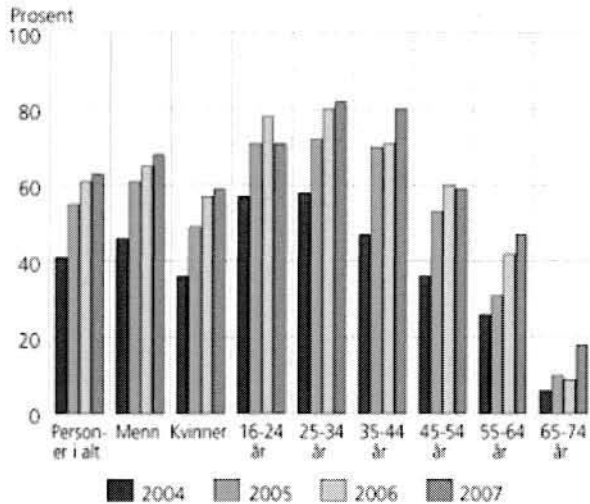
Ut fra vår situasjon, hvor vi er en produkt/tjenestetilbyder som videreformidler gjennom partnernettsverk som igjen gir oss tilgang til en rekke anerkjente merkeleverandører vil ikke vårt hovedfokus bestå av å kartlegge markedet for de produktene våre leverandører tilbyr. Våre produktleverandører er kjente, veletablerte aktører i markedet. Vi ønsker derved ikke å bruke mye ressurser på markedsundersøkelser i denne fasen av prosjektet. For oss ligger de viktigste suksesskriterier i markedskartlegging som tilskaffer oss kunnskap om våre hovedkonkurrenter og forbrukerens internettvaner hva angår netthandel.

Vi finner det naturlig å sammenligne oss med aktører som vi vil konkurrere med. Identifiserte konkurrenter er www.finn.no, www.forbruker.no, www.kelkoo.no, www.yahoo.no (Yahoo shopping), www.alatest.no, www.dinpris.no.

3.5.2. Bransjetall og trender

Basert på sekundærdata fra kildene; Statens institutt for forbruksforskning (SIFO) og TNS – Gallup har vi utledet følgende bransjetall og trender.

Nordmenns bruk av internett: Andelen av befolkningen som benytter internett til kjøp av varer og tjenester er sterk økende. I følge tall fra TNS gallup(2008) har antall daglige nettbrukere økt fra ca 1 million daglig rund 1000 årsskiftet til vel 28640000 pr. mars inneværende år. I figur 6 (SSB, 2007) som viser en grafisk fremstilling av statistikk fra statens institutt for forbruksforskning(2008) viser en markant økning i eldre nettbrukere som handler på via internett. Modellen viser andel av befolkningen som i løpet av 12 måneder i perioden 2004 til 2007 har handlet på internett til privat formål. Totalt omsetning for internetthandel i Norge er over 30 milliarder årlig.



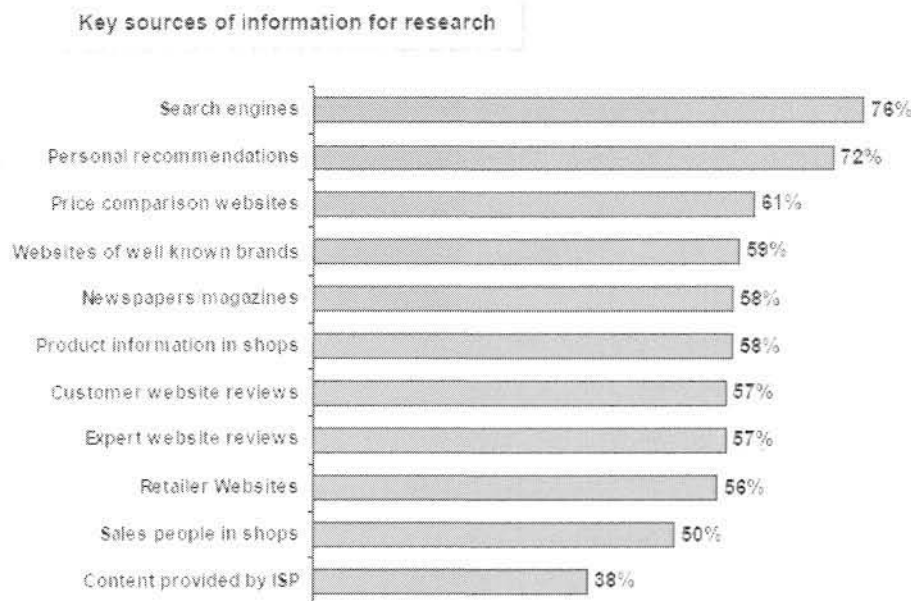
Figur 6 – Grafisk fremstilling over utvikling i netthandel.

En slik økning av godt voksne brukere forsterker vår tro på at enkelhet i brukergrensesnittet tilskaffer oss fortrinn i forhold til våre konkurrenter.

European Interactive Advertising Association (EIAA) har nylig kommet ut med en interessant rapport (EIAA Online Shoppers, 2008). Her kommer det frem en rekke interessante tall om bruk av Internett i Europa, og Nordmenn kommer frem som de mest aktive netthandlerne. Viktig å merke seg er at 44 % av alle nordmenn ombestemmer seg for hvilket merke de skal kjøpe etter å ha gjort undersøkelser på nett. Viktigste kilde for undersøkelser på nett er søkemotorer som oppgis til å være hele 76 %. På plassene etter dette følger personlige anbefalinger og nettsteder for prissammenligning. Nordmenn handlet i gjennomsnitt for 1203 Euro per person eller rundt 10.000 kr på nett i fjor. Dette er mest av alle landene som var med i undersøkelsen.

Videre sier rapporten at 67 % av alle norske nettbrukere handler på nett. Dette er også mest i Europa.

Som den grafiske fremstillingen av dataene viser i figur 7 (EIAA Online Shoppers, 2008) kan en se hvilke parameter som scorer høyest blant de norske brukerne.



Figur 7 – Norske netthandlers valg for å finne kilden til varen de vil kjøpe

Modellen gir oss gode indikatorer på hvor vi bør sette inn ressurser med henblikk på å tilegne seg mer kunnskap om brukernes vaner, hva de liker og hvordan de opptrer. Vi ser også at prissammenligningssider slik som det vi selv sorterer under scorer høyt blant brukernes foretrukne valg når de skal vurdere/handle på internett. Også personlige anbefalinger scorer høyt noe som gir som styrer våre tanker inn mot tematisering og målrettet oppbygging av temasider sortert under våre respektive kategorier. Effekten av å prioritere disse forholdene optimaliseres kontinuerlig ved å holde fokus i vårt markedsstrategiske tiltak.

Undersøkelser utført av statens institutt for forbruksforskning (SIFO) utgav i 2006 en rapport å vedrørende forbrukerkompetanse. Her var det gjort multivariate undersøkelser for å avdekke hvem er gode eller dårlig på hva. Ut fra rapporten kan en lese til dels store forskjeller i befolkningen hva angår kompetanse i forbindelse med netthandel. Faktorene alder, utdanning, inntektsnivå har de absolutt største forskjellene. Kjønn og sivil status ikke så store forskjeller.. De funnene vi har gjort gir oss gode indikatorer som gir oss kunnskap om hvor ressurser bør settes inn Tilgjengelighetsgraden (synligheten) vi greier å skape i søkemotorer er svært viktig for oss. Det er her mesteparten av våre potensielle kunder søker etter informasjon når de skal handle på internett.

Det kan også være stor sannsynlighet for at brukeren ombestemmer seg, hvilket sier oss at må stå sentralt for oss å frem budskapet i når brukeren søker på noe. Prissammenligningstjenester og gode søketjenester hjelper våre brukere å finne riktig leverandør ved å sammenligne på pris.



Mange nettbutikker ønsker å knytte seg til slike tjenester som vi tilbyr, nettopp av den grunn at de vet brukeren ønsker å skaffe seg en oversikt over leverandører, priser og tilgjengelighet. Vi mener at applikasjonens brukervennlighet kan treffe godt i markedet.

3.5.3. Søkermotoroptimalisering, et viktig markedsstrategisk tiltak

Man kan godt si at søkemotorer er nøkkelen til at du og jeg skal kunne operere på internett. Rindova og Kotha (2001) beskriver hvordan Yahoo og Exite startet som studentprosjekt ved Stanford Universitet. Yahoo ble i starten 1993, bygd opp i et kataloghierarki, finnes fortsatt på <http://dir.yahoo.com>. Yahoo fusjonerte etter hvert med Alltheweb og Overture og startet i 2001 søkemotorer basert på bruk av *crawlers*. Studentene som startet Exite benyttet softwareteknologi med *crawlers* allerede i 1994. Disses funksjon var å følge linkene og således indeksere sidene etter gitte parameter og etter avanserte algoritmer. I dag er Google markedsleder blant søkemotorsidene. Google henter sin informasjon forskjellige steder også fra "menneskeskapt" databaser. Et eksempel er <http://www.dmoz.org/> verdens største menneskelige redigerte katalog etablert i opensource ånd hvor alt er gratis. Derfor et godt tips å registrere siden også der. Når man ser at Google har en markedsandel i Norge på 80 – 90 % og rapporten fra EIAA (EIAA Online Shoppers, 2008) sier at 76 % av de norske netthandlerne ikke går rett til kilden men benytter søk gjennom søkemotorer (gjærne Google) ser man hvor viktig kunnskap om, og ressursinnsats på søkermotoroptimalisering er. Derfor er dette et fokusområde i alle deler av vårt strategitriangel. Effektiv søkermotoroptimalisering betinger at alle områder i vårt strategitriangel har et bevist forhold til gjennomføringen av arbeidet.

Vi kan dele optimaliseringsarbeidet inn i 2 hoveddeler, organiske treff og betalte treff

3.5.3.1. Organiske treff

Ombefatter de besøkende vi får inn på siden ved at vår side er listet i den søkemotor de benytter. Ettersom vi tidligere har påpekt at Google er markedsleder er det viktig for oss å indeksere siden vår hos dem. Google tilbyr gratis pakke med verktøy for optimalisering av indekseringsarbeidet. Nøyaktighet vil generere mer trafikk og denne trafikken er gratis. Søkermotoroptimalisering består av teknisk og kvalitetsmessig tilretteleggig av rangeringskriterier i søkemotorene.

Noen av de viktigste positive forhold er nøkkelord i titteltag, ankertekst i linker, alder på domene, interne linker, eksterne linker - link popularity.



Viktige ting en må unngå er server ikke tilgjengelig for crawlers, duplikat av sideindex, duplikat av innhold, duplikat av tittelags og nøkkelord, lav kvalitet og liten relevans på eksterne linker.

Forbrukertipsets applikasjon autogenererer innhold, slik det er i dag er dette en svakhet. Opprettelse av sitemaps og tematiske undersider, sortert jamfør våre kategorier er nødvendig for at søkemotorene skal kunne indeksere innhold fra oss. Begrensninger i antall nøkkelord gjør også sitt til at dette er påkrevd fordelt på flere undersider. Kvalitet i arbeidet her vil bidra til mulighet for mer målrettet annonsering inn mot undersider, eller landingssider for våre annonsekampanjer. Optimaliseringsarbeidet er en kontinuerlig prosess, resultater evalueres fortløpende via statistikkverktøy som google analytics⁶. Viktige forhold er valg av teknisk plattform, teknologiområdet må svare ut med å benytte standarder som er tilrettelagt for søkermotoroptimalisering. Brukerområdet må også svare ut på forhold som påvirker rangering i søkermotor i positiv retning.

3.5.3.2. Betalt trafikk

Fortsatt er det mange som benytter bannerannonser, betalt reklame på andre nettsteder, for å generere trafikk. Dette er risikofyllt med liten forutsigbarhet for hva man får igjen for annonsekostnadene. Videre er det svært mange som vil selge oss oppfølginger i diverse nettsteder, Nettkatalogen, Gulesider, Kvasir med videre. Erfaringen så langt er at dette er dyrt og lite forutsigbart. Pågående annonseselgere har lite dokumenterte opplysninger om forventet trafikk, i alle fall svært lite om hvilken CTR man kan forvente.

I senere år er det blitt svært populært å benytte betalte annonser tekstannonser hvor en betaler pr klikk - PPC.

Disse annonsene kommer opp når brukeren gjør søk i søkemotorer. Størst på markedet er Google med sin løsning Ad Words⁷, deretter Yahoo med sin Sponsored Search⁸ og MSN med sin lignende Live Search.⁹ Alle 3 har samarbeidspartnere slik at annonsene profileres på mange steder. Felles for alle er at vi som brukere har kontroll med kostnadene gjennom å sette tak på daglig budsjett.

⁶ <http://www.google.com/analytics/no-NO/index.html> Verktøy for optimalisering.

⁷ <https://adwords.google.com/select/Login> Plattform for annonsehåndtering hos Google

⁸ http://searchmarketing.yahoo.com/no_NO/arp/srch.php?o=NO0006 Yahoo sin løsning for annonsehåndtering.

⁹ <http://advertising.microsoft.com/norge/startsidene??PS=70635&NC=10009&CE=45&CP=1252&HL=Annonser+p%E5+MSN> Msn sin tekniske løsning for annonsehåndtering.



Prisen settes av oss selv etter hva vi er villig til å betale pr klikk – PPC, og er slik at den som byr mest på nøkkelordene som ønskes gjort søkbare rangeres høyest, det vil si den som betaler mest får den beste plasseringen. Løsningene har avanserte oppfølgingsmuligheter via statistikk, oppfølging av annonsekampanjer for optimalisering av konverteringsraten. Denne raten måler hvor mange prosent av klikkene på en annonse som ender opp med en transaksjon og er derved meget viktig i forhold til investere kroner i markedsføring. Google analytics har funksjonalitet som kan ivareta oppfølging av annonsekampanjer på flere plattformer. En kan for eksempel kjøre en adwords kampanje opp mot googles eget annonsesystem, samtidig kan en hente inn statistikk fra andre kampanjer. For eksempel banneroppføringer etc. Softwaren ivaretar beregning av konverteringsrater og på denne måten kan en estimere hva som vil være optimal anvendelse av annonsekronene.

3.5.3.3. Hvordan optimalisere betalt trafikk for best mulig return of investment. [ROI]

Vi benytter begrepet *return of investment* knyttet opp mot markedsføring for å belyse hvor stor avkastning hver investerte krone gir. Fokus på dette er å betrakte som de mest kostnadseffektive og påvirkbare tiltak vi kan gjøre. Med påvirkbare tiltak mener de forhold som kan være med på å påvirke vår inntjeningssevne i det arbeidet som gjøres. Faktorer som optimalisering av landingssider, visuell uttrykksform og tekstbruk er viktige. Boka *webdesign for ROI* (Loveday & Niehaus, 2008) betegner forholdet som en absolutt suksessfaktor.

Forretningsområdet og klientsideområdet må kontinuerlig samvirke for optimalisering av annonsekampanjer. Tidligere omtalte CTR (Click To Rate) er den prosentandelen av besøkende som klikker på annonsen eller linken vår om det er organiske treff.

Videre er konverteringsraten er den andel av dem som klikker som faktisk ender opp med en transaksjon. Som indikator på viktigheten illustrert i tabell XX kan en se at økning i konverteringsraten på så lite som en 1/2 % i dette eksemplet gir en dobling av inntekten.

Hvis en da tar utgangspunkt følgende: Tabellen illustrerer en tid på en måned. Det dårlige resultatet i kolonnen ”før” i modellen viser at vi tjener kr 24.000 denne måneden ved å kjøpe trafikk for kr 50.000. Konverteringsraten er lav bare 1,5 %. Dette gir oss en engangsutbetaling med fortjeneste på 24.000. Påfølgende måned påløper nye kr 50.000 for å generere tilsvarende inntekt. På et år gir dette $kr\ 24.000 * 12 = 288.000$

Modellen beviser at om vi setter inn kr 50.000 kronene på optimalisering rettet mot målrettet innhold, målrettete landingssider for annonsekampanjene og at vi ved dette arbeide øker konverteringen med en 1/2 % gir dette følgende regnestykke. Måned 1 har vi brukt pengene på optimalisering så dette generer ingen inntekt, isolert sett.



Imidlertid kan en se at når 50.000 settes inn i måned 2 og de neste 11 månedene sitter man igjen med følgende regnestykke.

jamfør tabellkolonne ”etter”: $49.000 * 11 = 539.000$

Illustrasjon av konverteringsrater og return of investment			
		Før	Etter
A	Antall Besøkende	50 000	50 000
B	Kostnad per Besøk	kr 1	kr 1
C	Kostnad for alle Besøkende (A*B)	kr 50 000	kr 50 000
D	Konverteringsrate	1,50 %	2,00 %
E	Konverteringer (D*A)	750	1 000
F	Inntekt per Konvertering	kr 100	kr 100
G	Totalinntekt (F*E)	kr 75 000	kr 100 000
H	Andre kostnader i %	50 %	50 %
I	Andre kostnader (H*G)	kr 1 000	kr 1 000
J	Markedsføringskostnader (C)	kr 50 000	kr 50 000
K	Profitt Totalt (G-[I+J])	kr 24 000	kr 49 000
L	Total ROI (K/C)	48,0 %	98,0 %
ROI forskjell			204,2 %

Tabell 1 - Konvertering og Return of investment

Forholdet påvirkes gjennom beviste valg, kvalitet i alle ledd. Alt arbeid med godt håndverk i søkermotoroptimaliseringen bidrar positivt. Landingssider for annonsekampanjer, enkelhet, tydelighet i budskapet, bevist forhold til usability, visibility.

Vårt nettsted må utvides slik at vi får tematiske landingssider, en for hver kategori fordi uten dette vil våre annonsekampanjer ikke treffe brukeren. Eksempel brukeren klikker på en link til oss etter å ha søkt på flatskjerm bør brukeren ikke komme til startsidene men til landingssiden for kategori lyd og bilde.

Tester kan gjøres som A / B tester inn mot sammen annonseleverandør, samtidig bør det testes leverandører imellom ved å kjøre identiske annonser samtidig hos Google, Yahoo og Msn. A / B tester, det vil si 2 identiske annonser med forskjellige landingssider for å teste ut hva som gir best konvertering – her kan små endringer vær nok – tekstbruk – visuelt førsteinntrykk. Overvåkning og analyser av all trafikk og annonsekampanjer spesielt er viktig. Hvor lenge er brukeren inne på siden(e), hvor svikter det, hvor kommer trafikken fra og en mengde parameter som gir nyttig informasjon om brukerens atferd.



Bevist forhold til hvor mye vi er villig til å betale for trafikk, statistikk vil hjelpe oss med å finne typiske konverteringsrater for de forskjellige kampanjene våre. Dette er upløyd mark for vårt prosjekt så tett kostnadsoppfølging (daglig) er påkrevd for optimalisering.

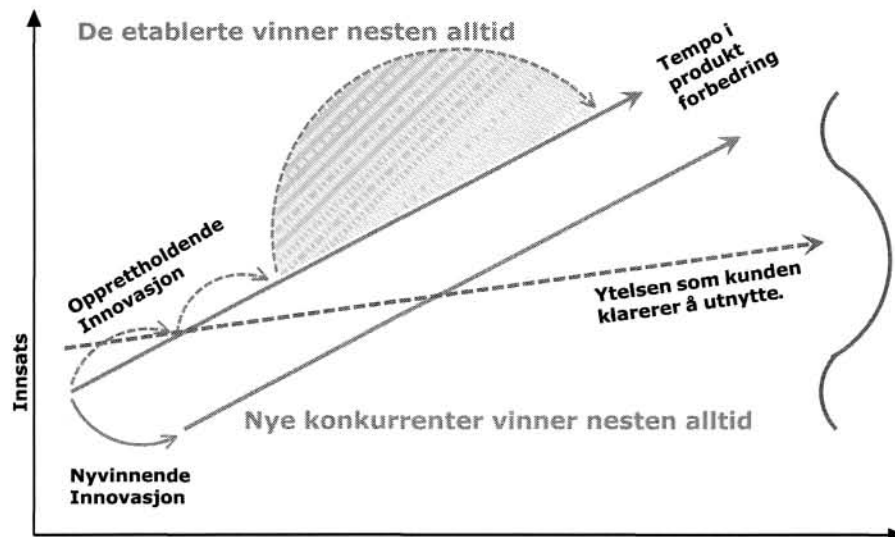
Viktig å huske er at all trafikk vi selv greier å generere gjennom søkermotoroptimalisering er gratis og kommer i tillegg til det som genereres via betalt markedsarbeid.

3.5.4. Hvilken strategi kan vi så treffe vårt marked best med.

Mihaeal E. Porters "five forcers" modell, Fernstrøm et al. (2005: 106), er et naturlig utgangspunkt for vurdering av de forhold som kan påvirke vår forretning. Porter påpeker at i tillegg til konkurranse mellom eksisterende aktører i vårt marked forhold i omgivelsene som kan påvirke enhver bedrift. Slike forhold er klassifisert i fire grupper: kunder, leverandører, potensielle nye inntrengere på markedet og trusler fra nye produkter eller tjenester som kan utfordre eksisterende løsning. Porters teorier betinger nøyaktige markedsanalyser og nøyaktig strategisk arbeid.

Det faktum at internett og de tjenester som tilbyes er i en rivende utvikling betinger kanskje at en må tenke annerledes enn før. Det er rett å slett ikke tid til langsiktig tradisjonell strategitenking. Av denne grunn har vi latt oss inspirere av Clayton M. Cristensen med sin noe annerledes tenkning i forhold til strategier og innovasjoner (Clayton, 2005). Clayton snakker om disruptive innovasjoner og benytter også uttrykk som disruptive teknologier. Ordet disruptiv direkte oversatt betyr splittede, oppløsende, undergravende.

Clayton skiller mellom det tradisjonelle som betegnes som en substaining innovasjon som baserer seg på tradisjonell teori med fokus på videreutvikling av produkter, tjenester og forretningsmodeller og det som han kaller disruptive innovasjoner som bryter med etablerte mønstre.



Figur 8 – Illustrasjon av substandig innovasjon og disruptiv innovasjon

Disruptive innovasjoner kan benyttes innen mange dimensjoner. Eksempler er dimensjonene kunder, forretningsmodeller, teknologi, konkurrentenes reaksjonsmønstre og strategiske tilpassninger. Figur 8 illustrerer hvordan en disruptiv (nyvinnende) innovasjon treffer markedet i forhold til en substanding (oppretholdene) innovasjon. Som en ser ligger den grønne linjen som indikerer en nyvinnende innovasjon lavere enn en opprettholdene innovasjon. Den røde bølgede vertikale linjen til høyre indikerer hva markedet er modent for å utnytte.

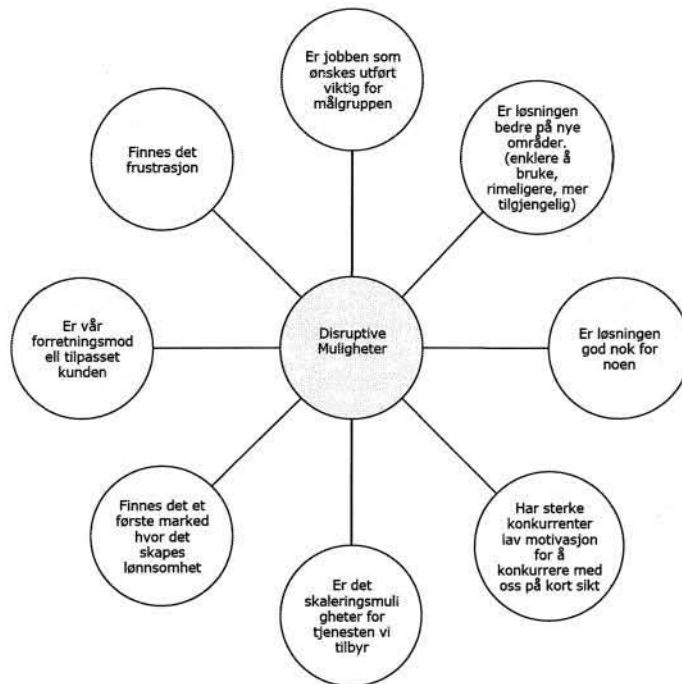
Vår motivasjon til å benytte Claytons teori som inspirasjonskilde er at den bygger opp under vår strategi om å benytte enkelhet i søketjenesten vi etablerer. Et eksempel kan være påståtte myter som: det koster for mye penger, det er vanskelig og lære og så videre. Det faktum at det for oss er mulig å etablere de ønskede tjenester vår forretningside skal levere gjennom bruk tilgjengelige gratis teknologi holder kostnadene nede. Det igjen viser at en etablering kan gjennomføres med begrenset tilgang på kapital. Web2 services basert på XML og tilgjengelige standarder åpner for nyetableringer uten stor kapitaltilgang. Utvikling av nye forretningsmodeller som baserer seg på nye verdikjeder gjør oss i stand til å bli innovatører i et marked tidligere dominert av kapitalstreke aktører.

Et annet viktig parameter er å sette seg i stand til å forstå kunden.

Clayton peker videre på åtte sentrale punkter, figur 8 som kan forsterke vår mulighet til å lykkes med vår internettsatsning. Ut fra de funn vi har gjort i forbindelse med trender, forbrukeradferd mener vi å langt på vei å oppfylle de 8 punktene.

At vi kan svare ut de 8 punktene med positivt svar er for oss ytterlige bekreftelse på at teorien passer for vårt bruk.

Vi utreder ikke de 8 punktene i modellen i detalj i denne rapporten, men tar argumentene med videre i prosjektet for ytterlige bearbeidelse.



t

Figur 9 – Åtte sentrale parameter i forbindelse med disruptive innovasjoner

3.6. Hvordan definere lønnsomhet - Innsats, tid og fortjeneste?

Vi kan sortere investeringsprosjekt i 4 hovedkategorier, investeringer som har til hensikt å erstatte noe som er etablert, investeringer som har til hensikt å øke kapasitet, investeringer som har til hensikt å etablere ny virksomhet og investeringer som har til hensikt å bedre faktorer i det indre miljø i bedriften. Vår investering faller inn under etablere ny virksomhet. Det finnes en rekke metoder for å foreta lønnsomhetsanalyser av investeringsprosjekter. De mest vanlige er (Westhagen et al. , 2004) Paybackmetoden er den enkleste som kun setter opp kontantstrøm(mer) med inn og utbetalinger uten å ta hensyn til faktorer som tidsverdien av kontantstrømmen(e). Annuitetsmetoden kan benyttes men er ikke så vanlig å benytte i prosjekter. En bedre metode er Nåverdimetoden, denne tar hensyn til tidsverdien siden nåverdien er det absolutte mål på lønnsomhet uttrykt i kroner.



Metoden ansees som å være den beste og mest brukte metoden.

$$NNV = \sum_{i=1}^n \frac{\text{verdier}_i}{(1 + \text{rente})^i}$$

Formel 1 - Utreking av Nåverdi på investeringen

3.6.1. Verdisette innsats

Kostnader forbundet med utviklingen av forretningsideen er i all hovedsak bundet opp mot bruk av egen tid, samt kostnader knyttet til leie av serverplass. Tidsforbruket frem til innlevering av bacheloroppgaven belastes ikke prosjektet. Vi prosjekterer med lansering av nettstedet den 1. september 2008. Bruk av egen tid belastes med kr 200 pr påløpte time i prosjektet fra tidspunkt for innlevering av oppgaven frem til årsskiftet og det tas ikke høyde for uttak av lønn eller annen provisjon i 2008. Det budsjettes med 10 timer ukentlig pr mann. Dette gir en samlet investering på ca kr 185000.00.

Totalt investerte midler frem til lanseringstidspunkt 1 jan. 2009					
Personell		Timepris	Timer/uke	Uker	Sum
Person 1	kr	(200,00)	10,00	30,00	kr (60 000,00)
Person 2	kr	(200,00)	10,00	30,00	kr (60 000,00)
Person 3	kr	(200,00)	10,00	30,00	kr (60 000,00)
Timekostnader					kr (180 000,00)
Hardvare, software, webhotell etc.					kr (5 000,00)
Totale kostnader					kr (185 000,00)

Tabell 2 – Totalt investerte midler fram til lanseringstidspunkt 1 jan 2009

Fra og med regnskapsåret 2009 er det satt opp en kontantstrømsoppstilling som illustrerer et budsjett med estimerte inn og utbetalinger. Oppsettet som er gjort i excel inkluderer NNV formel som beregner nåverdien av investeringen. Nåverdi > 0 indikerer ikke lønnsom investering, nåverdi < 0 indikerer lønnsom investering. Bergningen baserer seg på en kalkulasjonsrente på 8 % som igjen gjenspeiler tidsverdien på kapitalen. Renten kan også benyttes som tenkt avkastning på annen investering eller plassering av midlene som knyttes opp i prosjektet.

Budsjettet / kontantstrømsoppstillingen baserer påløpte egne ressurser og tall hentet fra øvelsen som er fortatt i punkt 3.5.4.3 jfr. tabell 1.



Faste kostnader og markedsføringskostnader er hentet fra tabell 1 i tillegg er det tatt høyde for uttak av lønn 10 timer a kr 200 ukentlig pr mann fra 1.1 2009 Budsjetteringen er ikke tilstrekkelig kvalitetssikret med henblikk på beregning av forventet salgsinntekter men mer ment som en illustrasjon av metodikken og den effekten fokus på return of investment kan gi en internettsatsning som vårt prosjekt. Som det fremgår av oppstillingen baserer tabell 3 seg på en konverteringsrate på 1,5 %. Dette ser vi kommer ut med en negativ nåverdi > 0 . Dette vil da si at prosjektet ikke er lønnsomt. Internrenten er negativ.

Busjett fra lansering 1 jan. 2009 fram til 31 des. 2011 basert på konverteringsrate 1,5 % jfr Tabell 1			
Benevnelse	År 2009	År 2010	År 2011
Salgsinntekter	kr 900 000,00	kr 900 000,00	kr 900 000,00
Lønn person 1	(kr 104 000,00)	(kr 104 000,00)	(kr 104 000,00)
Lønn person 2	(kr 104 000,00)	(kr 104 000,00)	(kr 104 000,00)
Lønn person 3	(kr 104 000,00)	(kr 104 000,00)	(kr 104 000,00)
Faste kostnader	(kr 10 000,00)	(kr 10 000,00)	(kr 10 000,00)
Markedsføring	(kr 600 000,00)	(kr 600 000,00)	(kr 600 000,00)
Netto kontantsrøm	(kr 22 000,00)	(kr 22 000,00)	(kr 22 000,00)
8 % Årlig diskontosats, kan representere inflasjonsrenten eller rentesatsen på annen investering			
kr (185 000,00)	Investeringskostnad basert på alternativkost		
(kr 22 000,00)	Netto kontantstrøm [utbetaling] i 2009		
(kr 22 000,00)	Netto kontantstrøm [utbetaling] i 2010		
(kr 22 000,00)	Netto kontantstrøm [utbetaling] i 2011		
NNV	Beskrivelse (resultat):		
kr (38 792,72)	Netto nåverdi av investeringen		
	Internrente	Negativ	

Tabell 3 – Budsjett fra lansering 1. januar 2009, basert på konverteringsrate på 1,5 %



Samme oppsett i tabell 4 hvor konverteringsraten økt til 2 %. Vi kan nå se at prosjektets nåverdi er < 0 , og er økt svært mye på grunn av en økning på en $\frac{1}{2}$ % i konverteringen. Forklaringen i dette er belyst i punkt 3.5.4.3. Internrenten er i dette eksemplet på 139 %. Den høye internrenten begrunnes med høyere konvertering, i tillegg er uttak av lønn satt lavt i prosjektperioden.

Busjett fra lansering 1 jan. 2009 fram til 31 des. 2011 basert på konverteringsrate 2 % jfr Tabell 1			
Benevnelse	År 2009	År 2010	År 2011
Salgsinntekter	kr 1 200 000,00	kr 1 200 000,00	kr 1 200 000,00
Lønn person 1	(kr 104 000,00)	(kr 104 000,00)	(kr 104 000,00)
Lønn person 2	(kr 104 000,00)	(kr 104 000,00)	(kr 104 000,00)
Lønn person 3	(kr 104 000,00)	(kr 104 000,00)	(kr 104 000,00)
Faste kostnader	(kr 10 000,00)	(kr 10 000,00)	(kr 10 000,00)
Markedsføring	(kr 600 000,00)	(kr 600 000,00)	(kr 600 000,00)
Netto kontantsrøm	kr 278 000,00	kr 278 000,00	kr 278 000,00
8 % Årlig diskontosats, kan representere inflasjonsrenten eller rentesatsen på annen investering			
kr (185 000,00)	Investeringskostnad basert på alternativkost		
kr 278 000,00	Netto kontantstrøm [utbetaling] i 2009		
kr 278 000,00	Netto kontantstrøm [utbetaling] i 2010		
kr 278 000,00	Netto kontantstrøm [utbetaling] i 2011		
NNV	Beskrivelse (resultat):		
kr 677 067,56	Netto nåverdi av investeringen		
	Internrente	139 %	

Tabell 4 - Budsjett fra lansering 1. januar 2009, basert på konverteringsrate på 1,5 %

3.6.2. Break Even

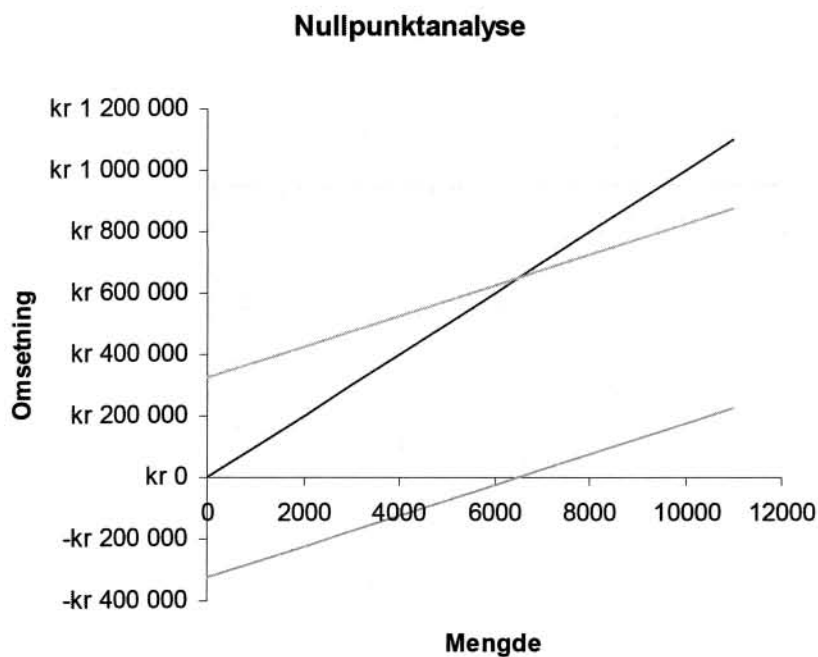
Nullpunktomsetning er regnet ut fra tall i tabell 1 og tabell 4 med følgende forutsetning. Ettersom vår forretningsmodell baserer seg på avtaler gjennom partnernettverk har vi selv ingen fysisk tilvirking av produktet. Vår definisjon på et produkt i budsjettmessig sammenheng vil bli at de 100 kr. vi har estimert som en gjennomsnittlig inntekt pr konvertering. Dette blir å betrakte som en produktenhet.



Videre vil utregning av dekningsbidrag og dekningsgraden basere seg på tall i budsjettet i tabell 4 hvor tallene for salgsinntekter og faste og variable kostnader igjen er hentet fra prognosen i tabell 1.

Type	Mengde	
Enhetspris	Kr 50	
Faste kostnader	Kr 322000	
Antall enheter	1200 stk	
Deltamengde	Tallet 1000	Tall for grafisk
DB pr enhet	Kr 50	
Dekningsgrad	50 %	
Nullpunktomsetning	644000	
Nullpunkt mengde	6440	
Total omsetning	Kr 1200000	
Risikomargin i kr	Kr 556000	
Risikomargin i %	46,33 %	
DB Tot	Kr 600000	

Tabell 5 – Økonomiske opplysninger for nullpunktberging



Figur 10 – Grafisk fremstilling av nullpunktomsetning i kr og antall.

Ut fra tallene fra nullpunktanalysen kan en gjøre noen betraktninger:

Vårt utkast til salgsbudsjett er som tidligere nevnt ikke tilstrekkelig kvalitetssikret.



I tabell 4 kan en se at total omsetning var estimert til kr 1200000, basert på kr 100 pr solgte enhet må nettstedet selge 12000 enheter årlig eller ca. 32 enheter daglig. Videre kan en se av tallmateriale at sikkerhetsmarginen var ganske stor. Dette gir følgende svar på nullpunktanalysen. Omsetning i antall vil være 6440 enheter årlig noe som gjør at vi må selge ca 18 enheter daglig. Det vil naturligvis være flere faktorer som påvirker om vi kan nå disse målene. Vi vet ikke hvor mange organiske treff vi får og hvor mange konverteringer dette kan resultere i. Ikke før vi har fått testet ut nettsiden vet vi hva vi greier å oppnå i konverteringsrate.

Vi vet heller ikke hvilke produkter vi kommer til å treffe best med, dette må testes ut slik at annonsekampanjer kan bli målrettet på de produktgrupper vi selger mest av. I budsjettet er det ikke tatt høyde for stort uttak av lønn.

3.7. Etikk og internett

3.7.1 Innledning

Virksomhet over internett, og da kanskje spesielt forretningsrelatert virksomhet, har skapt nye muligheter men også nye utfordringer og etiske problemstillinger. Dette kan oppsummeres på følgende vis:

”Mye debatt har blitt viet personvern og etikk på internett. Det faktum at internett krysser landegrensener gjør disse spørsmålene veldig komplekse. Forskjellige kulturer, lover, vaner og uvaner fra mennesker i forskjellige land gjør at hver geografiske region utøver egen policy på internett veldig forskjellig fra hverandre. Som et resultat vil alle som håndterer nettsider utfordres ved navigering i dette grumsete vannet av etisk bruk av internett” (Keri E. Pearlson, Carol S. Saunders, 2006: 208).

3.7.2 Bruk av informasjonskapsler

Vår forretningsmodell benytter tekniske løsninger som sporer og lagrer brukerinformasjon i form av cookies. Dette er informasjonskapsler som lagres på brukerens datamaskin og kan blant annet inneholde informasjon om hva slags valg brukeren foretok seg sist gang han eller hun var på Forbrukertipset.no sine nettsider. Dette er altså ikke å anse som skadelig programvare, men et nyttig hjelpemiddel for utviklere slik at brukerpreferanser kan lagres. For eksempel så benyttes dette ofte i forbindelse med handlekurven hos nettbutikker, da for å holde kurven oppdatert så lenge prosessen pågår. Det blir når denne type teknologi brukes kreativt eller snarere misbrukes at det oppstår problematiske gråsoner. Dersom du lagrer informasjon om brukeren så er det ikke alltid påkrevd å informere brukeren om dette. At



handlekurven er oppdatert under hele handleprosessen er et eksempel på bruk av informasjonskapsler som ikke krever at nettstedet informerer om dette. Ved lagring av personinformasjon i form av navn, telefonnummer, adresse og epost vil det forventes av de fleste i dag at nettstedet har en policy om personvern som behandler dette. Vi benytter verken handlekurv eller lagring av personopplysninger i dag, men som en del av et partnernettverk tase det i bruk informasjonskapsler på en slik måte at det kan reises etisk spørsmål rundt bruken.

3.7.2.1 Etisk gråsoner i forretningsmodell

Vår forretningsmodell baserer seg på provisjon av solgte produkter som kan relateres til at kunden har gjort bestilling etter å ha kommet inn via vår internettløsning. Teknisk sett foregår dette ved at brukeren finner frem til ønsket produkt fra leverandør A på vårt nettsted. Når brukeren klikker på bestill-linken på vår internettside vil han eller hun overføres til internettsiden for leverandør A for det spesifikke produktet. Det vil samtidig lagres en informasjonskapsel på brukerens datamaskin som peker mot vår nettside.

Dersom brukeren foretar et kjøp der og da regnes provisjonen ut og Forbrukertipset.no blir kreditert for dette. Det som gjør at denne løsningen havner i en etisk gråsoner er at informasjonskapselen lagres hos brukeren i inntil 45 dager etter at brukeren gjorde sine undersøkelser men ikke kjøper varen. I praksis innebærer dette at vi kan motta provisjon dersom brukeren handler varen på leverandør A sitt nettsted i over en måned etter at brukeren sist var på våre nettsider, altså uavhengig om brukeren/kunden går via oss eller direkte til leverandøren sin side.

3.7.3 Vår holdning til denne gråsonen.

Dette representerer et problemsett som vi er en del av som medlem i partnernettverket, men som vi ikke kan styre og ha kontroll over da disse informasjonskapslene ikke er generert fra vår kildekode. Våre brukere/kunder blir heller ikke skadelidende av denne praksisen da de betaler samme sum for varen som de ellers ville gjort.

Så lenge vi ikke misbruker denne muligheten bevist med tekniske tilpassninger i vår applikasjon, hvilket hadde vært teknisk mulig, anser vi denne etiske gråsonen er grei å være klar over. Men vi vil understreke at den ikke er av en slik karakter at den bør påvirke vår forretningsdrift.



3.8. Oppsummering

Bearbeidelsen av de valgte problemstillingene som sorterer under forretningsområdet har vært en spennende prosess. Å se prosjektet forbrukertipset.no i et helhetsperspektiv, knytte opp mot de 2 andre fokusområdene har stilt store krav til oss. Generelt synes vi å ha identifisert en rekke forhold som er sentrale suksesskriterier videre i forretningsutviklingen.

Spesielt vil vi trekke frem de markedsstrategiske valg vi har gjort. Claytons teori om disruptive innovasjoner har gitt inspirasjon til videre bearbeidelse av problemstillingene.

Teori innen best mulig return of investment, satt i en internettsammenheng har tilført prosjektet nyttig kunnskap som vil hjelpe oss å optimalisere nettsiden sett i dette perspektivet.

Økonomiske øvelser med henblikk på forholdet return of investment har gitt oss kunnskap som bidrar til god payback på hver brukte annonsekroner.

Nytten i forhold til tilegnet kunnskap på søkemotoroptimalisering har lært oss mye om nødvendig samspill mellom forretning, bruker og teknologi.

Hovedproblemstilling med tilhørende underproblemstillinger omtales under konklusjoner, punkt 6.2 i rapporten.



4. Brukerområdet

4.1. Presisering av problemstilling

Innledningen skisserte følgende problem og underproblemstillinger dedikert til behandling under brukeroområdet:

- Oppfyller prototypen forretningsvisjonen sett fra et brukerperspektiv?
 - Skjønner brukerne vårt innovative produktfiltreringskonsept?
 - Følger den utviklede nettapplikasjonen etablerte brukerkonvensjoner?

Ønsket om en utredning rundt disse problemstillingene bunner i tre klare behov.

1. Fra forretningsområdets visjon er det slått fast; ” Forbrukertipset.no skal gjennom utvikling av innovative løsninger lage brukervennlige og intuitive webløsninger for sortering og søk i store mengder data”. For tidlig å kunne avdekke om vår oppfatning av visjonen bak applikasjonen støttes av uavhengige brukere var gjennomføring av en brukertest nødvendig så snart en kjørbar prototyp var på plass.
2. Brukertesting anså også for å være en god tilnærming for å avdekke andre konkrete svakheter i vår prototyp sett fra et reelt brukerperspektiv.
3. Til slutt var det ønskelig å etablere en forståelse av hvilke etablerte brukerkonvensjoner som eksisterer og som kunne fungere som en beste praksis for videreutvikling av applikasjonen med brukeren i fokus.

Brukerperspektivet vil således viderebehandles ut fra disse behovene, men først har vi valgt å presentere en kort historisk tilnærming til brukerperspektivet når det gjelder applikasjonsutvikling mot internett.

4.2. Tilnærming – Løs historikk rundt brukerperspektivs utvikling

Utviklingen av nettapplikasjoner har vært preget av kreative tilpassninger mellom tekniske muligheter, standarder og hva nettlesere(web browsere) har støttet.

HyperText Markup Language (HTML) har vært standarden for presentasjon av data i nettleseren fra starten. HTML håndterte enkle tekstbaserte nettsider godt, men behovet for mer avanserte løsninger både med tanke på presentasjon av innhold og layout medførte at utviklere tok i bruk kreative teknikker som ikke HTML var tiltenkt for, samt klientside scripting, eksempelvis JavaScript, for mer dynamikk.



Resultatet ble mer avanserte nettsteder, men også mer ustabile nettsider på grunn av de kreative tilpassingene til en utilstrekkelig standard.

Koden ble også unødvendig komplisert og vanskelig å vedlikeholde for større nettsteder, når innhold og layout ble produsert om en annen i samme kode. Utviklingen av tekniske hjelpemidler for svaksynte og blinde, slik at disse også kunne orientere seg på Internett, var og er svært avhengige av at web standardene blir fulgt i forbindelse med utarbeiding av nettsteder. Det har derfor vært vanskelig for denne gruppen brukere å navigere på nettstedene som er bygget på kreative tilnærminger for å gjøre nettstedene mer avanserte og forlokkende. For å kunne få utviklingen inn i standardiserte former var det derfor sterkt påkrevd med en egen standard for layout som kunne avlaste HTML standarden. Forløperen til denne standarden kom i 1994 og er blitt kjent som Cascading Style Sheets (CSS). CSS versjon 2 er i dag offisiell nettstandard. Selv om dette har vært et viktig skritt for å standardisere utviklingen av nettsteder, henger enda bruken av de kreative tilpassningene igjen. Dette skyldes at det er en svært omfattende jobb å kode om eksisterende nettsteder, bruk av CSS krever også at utviklere tilegner seg ny viten og CSS standarden har også hatt manglende støtte i populære nettlesere.

Et annet utviklingstrekk ved nettapplikasjoner har vært den stadige mer omfattende bruken av multimedia, animasjoner, lyd og video. Dette har blitt tatt i bruk i takt med den økte overføringskapasiteten på linjenettet. Dette har igjen skapt problemer for blinde og svaksynte siden dette ikke har vært støttet i nettlelere tilpasset denne gruppen, samt at det ikke var noen god praksis for tilrettelegging mot denne gruppen brukere blant utviklere.

Disse problemene, samt behovet for å kunne vise samme nettsider over forskjellige plattformer, var utgangspunktet for Web Accessibility Initiative (WAI). Dette er en av undergruppene til W3C og de jobber med retningslinjer for webapplikasjoner som er tilgjengelig for flest mulig brukere. De første offisielle retningslinjene; Web Content Accessibility Guidelines (WCAG) 1.0 hadde som hovedmålsetning og gjøre webutviklere bedre i stand til å utvikle nettsteder som fungerer godt for personer med hemninger.

Med hemninger kan forstås; Svekket- sansevne, intellektuelt nivå eller manglende erfaring med data som gjør det vanskelig å bruke ordinære nettjenester.

Men det var også en klar målsetning at nettsteder som følger disse spesifikasjonene også vil oppnå en meget god brukeropplevelse for normalbrukeren, uavhengig av teknisk plattform.

WCAG 1.0 er fremdeles gjeldende retningslinje fra WAI, men har vist seg å være utilstrekkelig når det kommer til webutvikling hvor mer avanserte utviklingsteknikker tas i bruk.

Eksempel på avansert utviklingsteknikk er Asynchronous JavaScript And XML (Ajax).

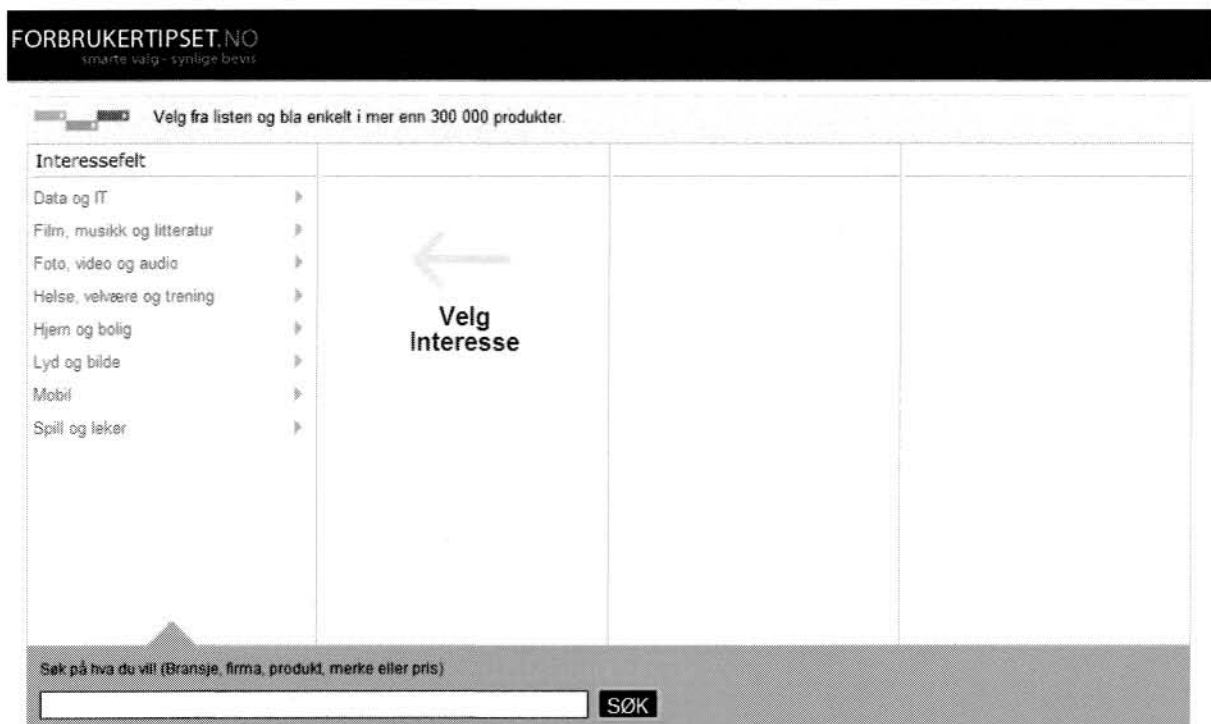
Denne teknikken baserer seg på kjente teknologier/standarder, men gir mulighet for kjappere datautveksling mellom klient og tjener. Bare nødvendige deler av siden oppdateres ved behov og datautvekslingen kan gjøres som en bakgrunnsprosess uten at brukeren merker det.

Bruken av avanserte teknikker sammen med behovet for mer presise testmetoder har ført til utarbeidelsen av WCAG 2.0 spesifikasjonene. Disse forventes ferdigstilt i løpet av 2008, men er allerede godt utarbeidet og er derfor anbefalt tatt i bruk av WAI.

4.2.1. Våre inngangsparametere med tanke på brukeren i prototyp utvikling.

Brukerperspektivet inn mot prototyp utviklingen var ikke mer utviklet enn at applikasjonen skulle løse visjonen redegjort for under forretningsområdet. Fokuset lå således i å teknisk generere kolonnene med tilhørende data og at disse skulle være enkle å navigere i. Det skal dog sies at det også er lagt til kolonneoverskrifter og pekere for å støtte brukeren.

Prototypen fremstår derfor visuelt sett nesten uten informasjon utover det selve produktsorteringskolonnene gir, se figur 11.



Figur 11 - Startsidens slik den fremsto ved brukertesting



4.3. Teste om visjon og konsept er løst i prototypen

Redegjørelsen for valg av design vedrørende brukertestene er gjort i kapittel 2 og vil således ikke berøres her.

4.3.1. Målsetninger, hypotese og begrensinger for brukertest av prototyp

Vår målsetning

Det vi ønsket å finne ut gjennom brukertester er om brukeren faktisk skjønner vårt sorteringskonsept uten noen ekstra forklaringer utover det prototypen visuelt forteller. Videre var det ønskelig å se hva de forskjellige brukerne erfarte og tenkte rundt den informasjonen som er tilgjengelig i hver kolonne. Vi var også spente på hvordan søkemotoren i bunnen av applikasjonen ble oppfattet da den er litt uortodoks plassert.

Forventet resultat - hypotese

Testingen ble gjort ut fra en hypotese om at det kolonnebaserte sorteringskonseptet ville enkelt la seg forstå og dermed bidra til å støtte opp om vår visjon rundt produktsorterings applikasjonen.

Men det var også av interesse å avdekke andre svakheter og mangler som informantene erfarte gjennom testing av alle kolonner, samt ved å måtte besvare de enkle kontrollspørsmålene som var utarbeidet til testen. En del svakheter var vi klar over selv, men det vil alltid være nyttig å få slike ting bekreftet av uavhengige og friske øyne.

Begrensinger ved prototypen som influerer på testmulighetene

Alt etter hva informanten gjør av valg i applikasjonen kan det forventes at noen kommer i knipe med å finne et spesifikt produkt. Dette på grunn av mangelfulle sorteringsmekaniseringer i kolonne 3. I tillegg er ikke søkemotorfunksjonaliteten utviklet på klientsiden og kan derfor ikke testes.



4.3.2. Redegjørelse for gjennomførte brukertester av prototyp

Informant 1: Kvinne 42 år

(Bruker internett noe, har også bestilt varer på nettet før, men ikke brukt lignende tjenester).

Informant skjønnte straks at det kunne gjøres valg i kolonne 1. Kolonne 2 og 3 gikk også veldig greit. Var missfornøyd med informasjonen i kolonne 4, ønsket mer informasjon om hvert produkt tilgjengelig her for kunne gjøre et kvalifisert valg. Syntes også kolonne 5 gav lite informasjon og reagerte på at produktbilde ikke så ut til å være i rett perspektiv. Fant ikke retur Stolpen, men prøvde nettleserens ”gå tilbake” funksjon når hun skulle starte på nytt. Hun la ikke merke til søkemotoren i bunnen av applikasjonen.

Generelt inntrykk:

- Negativt, først og fremst på grunn av manglende informasjon til å kunne gjøre et kvalifisert valg i kolonne 4. Var også misfornøyd med produktpresentasjonen i kolonne 5.

Informant 2: Kvinne 56 år

(Bruker internett noe, men ikke til produktkjøp)

I likhet med foregående informant skjønnte vedkommende straks at det skulle gjøres valg i kolonne 1. Hun hadde ingen problemer med å navigere gjennom alle kolonnene frem til et spesifikt produkt. Når hun skulle tilbake til interesseområdet forsøkte hun først nettleserens ”gå tilbake” funksjon, når denne ikke virket klikket hun på Forbrukertipset.no i hodet på nettsiden. Da ble siden oppdatert og kolonne 1 var tilgjengelig. Hun så ikke retur Stolpen i applikasjonen, men syntes denne var smart når hun ble gjort oppmerksom på den. Hun la heller ikke merke til søkemotoren i bunnen av applikasjonen. Når hun klikket på bestill i kolonne 5 skjønnte hun ikke at hun hadde forlatt vår nettside og kommet inn til en ny nettside. Informanten prøvde også å justere størrelsen på produktbildet i kolonne 5 ved å trykke på zoom knappen. Dette er bare et statisk bilde i applikasjonen og har ingen funksjonalitet og kan med fordel fjernes for funksjonalitet er innarbeidet.

Generelt inntrykk:

- Positivt, syntes det var en enkel og morsom måte å lete gjennom produkter på. Hun mente bestemt at søkeboksen måtte flyttes opp i hode på nettsiden hvis den skulle bli sett.

**Teknisk problemer identifisert i testprosessen:**

- Når alternativer ikke er tilgjengelig i en kolonne blir applikasjonen stående å søke uten stopp.
- Ved valg av noen kategorier kommer prisfiltrerings alternativ: "Maks Pris NOK:99000" opp uten at det linker til noen produkter.

Informant 3: Mann 42 år

(Bruker internett daglig, kjøpt produkter online flere ganger fra ulike leverandører)

Informanten forsto intuitivt hvordan navigasjonsprosessen foregikk. Han hadde ingen problemer med navigasjonen og fant raskt frem til ønsket produkt. Informanten forsøkte å navigere tilbake for å se på andre produkter, men fant ikke retur Stolpen. Informanten forsøkte derfor å bruke nettleserens tilbakeknapp og ble forvirret over da å bli tatt vekk fra siden. Han ga tilbakemelding om at produktdetaljene var for begrenset i kolonne 5 og mente bestemt at dette måtte bli bedre. Han så ikke søkeboksen og ble overrasket da han ble gjort oppmerksom på den. Tilbakemeldingen på dette var at den burde stått på oversiden av kolonnene og ikke på undersiden.

Generelt inntrykk:

- Mente det kunne være noe han ville bruke dersom påpekte ting ble gjort bedre. Bedre produktdetaljer, søkeboksen på oversiden av kolonnene og tydeligere tilbakelink til foregående kolonner.

Informant 4: Gutt 12 år

(Bruker internett, men ikke til produktkjøp)

Informanten skjønnte at han skulle gjøre valg i kolonnen, men skjønnte ikke hva vitsen med det skulle være. I kolonne 3 stoppet det helt opp for vedkommende skjønnte ikke hva som sto i kolonnen. Bruken av forkortelser bidro til dette, så når det ble oppklart var det greit. Når informanten fikk opp produkter i kolonne 4 skjønnte vedkommende hva bruksområdet for applikasjonen var. Han syntes det var lite informasjon i kolonne 5, det aktuelle produktet. Når han ble bedt om å returnere til første kolonne prøvde han via nettleserens "gå tilbake" funksjon. Da dette ikke lyktes viste han ikke hva han skulle gjøre. Han så ikke retur Stolpen i applikasjonen, ei heller hadde han fått med seg søkemotoren i bunnen av applikasjonen.

Generelt inntrykk:

- Litt vanskelig å skjønne hva dette var for noe med en gang, men at det var en morsom måte å lete opp produkter på når han viste at det var det som var poenget.



4.3.3. Konkrete ufordringer som funnene i brukertestene oss.

Brukertestene mener vi har styrket vår visjon rundt konseptet produktsortering i kolonner. Med unntak av en informant, 12 år gammel gutt, forsto alle hva dette gikk ut på med en gang. Den refererte gutten forsto ikke med engang hva målet med applikasjonen var, men forsto konseptet etter å ha gjort valg i kolonnene frem til kolonne 4.

Videre har testene avdekket et entydig behov for å flytte søkeboksen vekk fra bunnen av applikasjonen, da ingen av informantene så denne under testen. Det må antas at hvis informantene hadde følt et behov for å gjøre et direkte søk ville de nok fått øye på muligheten, men dette holder ikke for oss.

Det er også helt entydig slått fast gjennom testene at den navigasjonsmuligheten som er lagt inn som en stolpe til venstre i applikasjonen, etter at det er gjort valg i kolonnen ”produkter”, ikke legges merke til. Denne navigasjonsmuligheten ble bygget inn i applikasjonen for at brukeren skulle kunne få tilbake ”interesse” kolonnen som forsvinner med påpekte valg. Denne navigeringsmuligheten er svært viktig da det ikke går an å bruke funksjonaliteten som er innebyggt i nettleserne for dette formålet i dag.

4.4. Identifisering av brukerkonvensjoner og om disse følges i prototypen

4.4.1 Identifisering av grunnprinsipper

Krug (2006:s.11) mener at ”Ikke få meg til å tenke” er det overordnede prinsipp når det gjelder å vurdere brukervennligheten på et nettsted. Hva vil så dette innebære?

Jo at en bruker som havner på vårt nettsted må raskt og enkelt skjønne nettstedets funksjonalitet og hvordan vedkommende kan navigere seg videre på nettstedet.

Den vanligste måten en bruker havner på et nettsted i dag er via en søkemotor. Brukeren er gjerne på leting etter noe mer eller mindre spesifikk. Undersøkelser viser at denne leteprosessen består i å la blikket gli over en webside på jakt etter tekst som kan virke interessant i forhold til det brukeren er ute etter. Brukeren leser altså ikke all tekst, men skanner over siden i full fart og gjør seg opp en mening fortløpende.

Dette gir noen kritiske faktorer som må oppfylles for at brukeren skal vurdere å stoppe opp.

- Søkene som gir treff mot vår side via søkemotorene må stemme med hva siden vår tilbyr.
- Sidene på nettstedet må være ryddige og oversiktlige. De må enkelt og klart uttrykke hva brukeren kan få gjort der.
- Navigasjon videre på nettstedet/applikasjonen må være lett forståelig og godt synlig.



Når så brukeren har stoppet på en web siden fordi den virker tiltalende og ser ut til å dekke behovet, hva er det da som kan hjelpe brukeren i den videre leteprosessen etter sitt endelige svar? Følgende retningslinjer bidrar i stor grad til den hjelpen brukeren trenger.

- Skap et klart visuelt hierarki på hver side.
 - *Fontstørrelser, farger og rammer kan markere dette.*
- Bruk kjente konvensjoner hvis du ikke kan lage noen bedre selv.
 - *Dette gir en stor grad av gjenkjennbarhet for brukeren.*
- Del opp siden i klart definerte områder.
 - *Gjør det ryddigere for brukeren å fokusere på det han/hun er ute etter.*
- Få klart frem hva som er klikkbart på siden.
 - *Forteller brukeren hvor han/hun kan finne mer informasjon.*
- Reduser støyen på siden.
 - *Unngå unødvendig visuell støy på siden. Det være seg tekst, bilder, animasjoner, lyd.*

Tekstbruk på en webside er svært viktig for å kunne formidle et klart budskap. I tillegg er tekstbruken viktig for god indeksering hos søkemotorer. Men for mye tekst vil bidra til å skjule budskapet. Derfor bør tekstbruken reduseres til et absolutt nødvendig nivå. Dette vil ha flere positive effekter ovenfor brukeren, samt indeksering hos søkemotorer:

- Støyen på siden reduseres når unødvendig tekst kuttes.
- Den viktige teksten fremheves og budskapet kommer klarer frem.
- Websiden fremstår som kortere og ryddigere.

Prosessen med å fjerne og foredle tekst vil bidra til en bevisstgjøring for eier av websiden om hva som virkelig er viktig og nødvendig. Dette vil kreve en meget nøye overveielse i bruk av ord og dermed gjøre teksten enda mer treffsikker.

Navigasjon på et nettsted skal fylle mange funksjoner og uformingen av denne er derfor viktig med tanke på å hjelpe brukerne. Følgende positive funksjoner kan tilskrives navigasjonen.

- Hjelp brukeren å finne det han/hun leter etter.
- Hjelp brukeren å holde orden på hvor han/hun er på nettstedet.
- Hjelp brukeren å få oversikt over hva nettstedet inneholder.
- Hjelp brukeren å forstå hvordan nettstedet skal brukes.
- Skape tillit ovenfor brukeren om at disse er tenkt på ved utvikling av løsningen.

Det eksisterer navigasjonskonvensjoner for websider og en standard tilnærming kan ses i figur 12.



Figur 12 – Navigasjonskonvensjoner illustrert

Ordet navigasjonskonvensjon tilsier at dette er et oppsett som bør følges på internettsider hvis en ikke kan komme opp med noe bedre. Det skal sies at denne konvensjonen ikke gjelder for startsider. Disse må tilpasses slik at de dekker det informasjonsbehovet som nettstedet krever. Dette unntaket gir videre at også undersider til et nettsted kan unnvike fra konvensjonen hvis presentasjonsbehovet er totalt annerledes enn i gitt konvensjon. Grunnprinsippene for navigasjon vil dog være gjeldende.

4.4.2. Konkrete ufordringer opp mot brukerkonvensjoner

Prototypen som en selvstendig webside gir for lite informasjon om hva formålet med siden er. Videre har brukertestene avdekket at navigasjonen ikke er tilfredsstillende, dette fremheves som svært viktig for brukeropplevelsen i teoridelen. Søkeboksen som var plassert i bunnen av applikasjonen ble ikke sett av testkandidatene. Teorien bekrefter at dette er en plassering som bryter med klare konvensjoner og således støtter opp om funnene i brukertestene.



Vårt valg vedrørende kolonnestruktur for filtrering av produkter gir en sideutforming som ikke faller helt inn under standardkonvensjon for en webside. Dette kan forklares med at dataene som skal presenteres er av en slik art at standardkonvensjonen ikke passer. I tillegg er dette tenkt som en endring av webkonvensjoner for å forbedre slike internettjenester, derav vår visjon.

4.5. Oppsummering

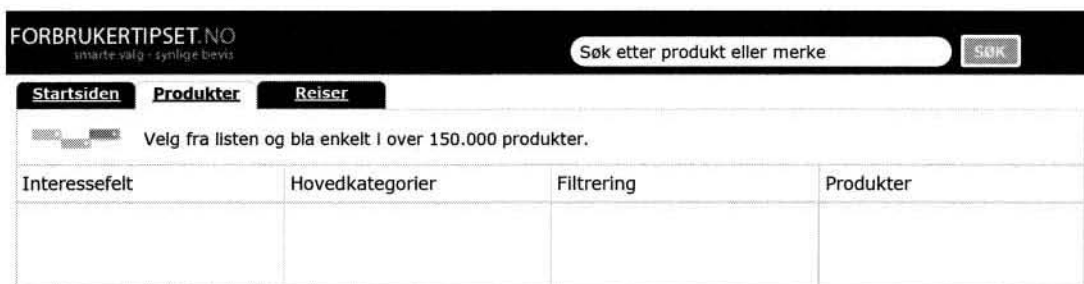
Testing av prototypen, selv på dette enkle grunnlag, har vært nyttig og bidratt til konkrete behov for endring av applikasjonen. Dette sammen med de generelle beste praksis trekkene fra relevant teori gir derfor følgende føringer for videre utvikling av applikasjonen.

Prototypen utgjør i dag en enkelt webside. I den konteksten er det litt for lite informasjon til brukere som kommer til denne siden. Videre utviklingsplaner for Forbrukertipset.no inneholder allerede planer om enda en tjeneste i tillegg til produktsøk, dette gjelder en egen reisebestillingsmodul som skal implementeres i nær fremtid. Dette betinger videre at det etableres en egen startside som kan presentere nettstedet Forbrukertipset.no.

I en slik kontekst trenger ikke produktfiltreringsapplikasjonen å tilføres noe mer enn en ekstra navigasjonsmulighet som kan ta brukerne mellom nettsidene på nettstedet.

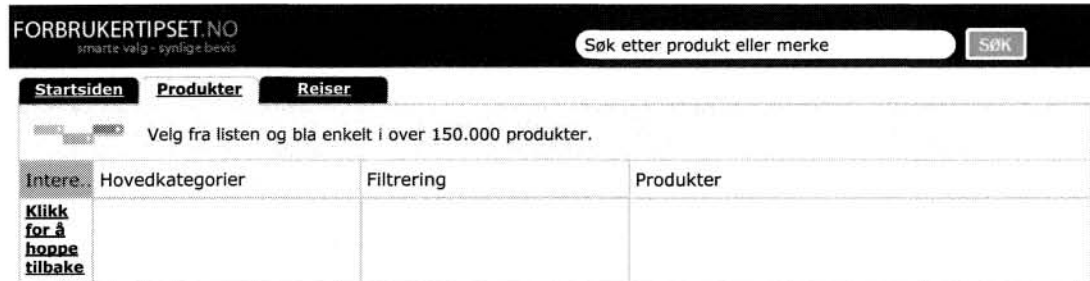
Informasjonen vedrørende nettstedet vil da bli ivaretatt på startsidene.

For å imøtekomme de konkrete svakhetene som ble avdekket under brukertestene må søkeboksen gjøres mer synlig i applikasjonen, samt at kolonnestrukturen må endres noe. Et tilpasset starbilde på produktfiltreringssiden, se figur 13, er da kolonnemessig som før, men har fått tilført en overordnet navigasjon og søkeboksen er nå plassert i hodet kontra bunnen. I tillegg er søkeknappen gjort mer fremtredene.



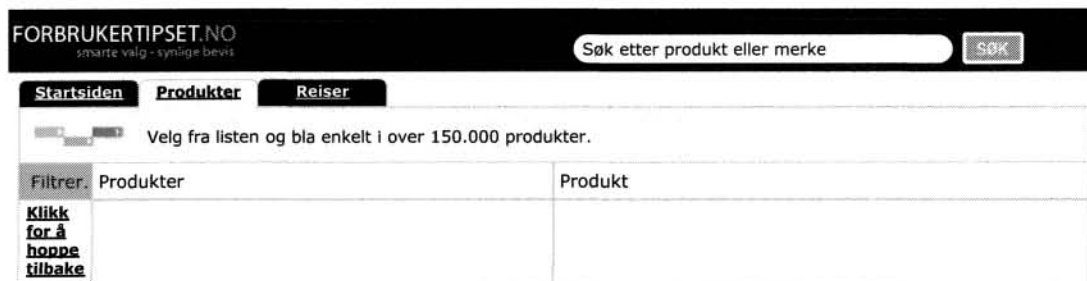
Figur 13 – Det nye produktsorteringsapplikasjonen

Når brukeren kommer så langt at han/hun gjør valg i filtreringskolonnen og produktene skal presenteres vil det være ønskelig med en større kolonnebredde for dette. Figur 14 viser hvordan dette kan se ut. Den første kolonnen er nå nesten skjult, men klart markert som en navigasjonsmulighet tilbake til startbilde som vist i figur 13.



Figur 14 – Det nye produktsorteringsapplikasjonen, første progresjon.

Når så brukeren bestemmer seg for et produkt vil det være ønskelig at produktet også kan presenteres i en større kolonne enn tidligere. Dette for å gjøre plass til mer produktdetaljer, se figur 15 for et eksempel på kolonnebredde. Kolonnene tilpasser seg derfor slik at bare to kolonner er direkte valgbare for brukeren nå. Dette gir brukeren den nødvendig fleksibiliteten til å raskt velge et annet produkt innen samme filtrering. I tillegg er navigasjonen tilbake til tidligere kolonneoppsett, som vist i figur 14, også her klart markert.



Figur 15 - Det nye produktsorteringsapplikasjonen, andre progresjon.

Bunnlinja i produktsorteringsapplikasjon er ikke tatt med her. Den bør erstattes med en navigasjonslinje som skal være gjennomgående for alle nettsidene på Forbrukertipset.no. Figur 12 gir en pekepinn på hva som kan inngå i en slik bunnlinje.



5. Teknologi området

5.1 Presisering av hovedproblemstilling

"Hvilke teknologiske og tekniske forutsetninger vil ligge til grunn for en realisering av applikasjonen?"

I vår strategitriangel er det et samspill mellom forretning, bruker og teknologiområdet. Det krever en balanse mellom disse områdene og dersom det er tekniske problemstillinger eller behov for teknologi som ikke lar seg realisere innenfor gitte rammer så må forretningssiden justere sitt område. Samtidig er det viktig å minne om at det er forretningssiden som har de overordnede strategiske målene og det er den som driver frem problemstillinger og utfordringer for teknologiområdet. På samme måte kan også brukerområdet avdekke problemstillinger som er av teknologisk og/eller teknisk art og derfor bringe nye behov til teknologiområdet. Hovedproblemstillingen er derfor en dynamisk problemstilling i den forstand at den er gjenstand for stadige endringer. Videre mener vi det er viktig å avgrense hva som faktisk ligger i selve formuleringen av problemstillingen. Vi vil derfor gjøre oppmerksom på at problemstillingen ikke er ment som et altomfattende teknologisk og teknisk spørsmål som skal besvares, men som en praktisk rettet utfordring innenfor den strategitriangelen vi har definert. Applikasjonen skal utvikles etter krav fra forretningssiden og på bakgrunn av brukerområdets arbeid vil/kan det oppstå flere definerte krav. Teknologiområdet vil behandles tredelt. Krav fra forretningsområdet, krav fra brukerområdet og utvikling av en prototyp. Vi skiller ikke spesielt på teknologiske aspekter og rent tekniske, men ser på dette under ett. Hovedvekten vil ligge på utvikling av prototypen da dette er den viktigste kilden til problemløsning og konklusjon. Vi har valgt å formulere overskriftene på disse punktene som spørsmål fordi dette er direkte underproblemstillinger til hovedproblemstillingen.

5.2 Hvordan løse andre strategiske momenter fra forretningsområdet?

Selve forretningsideen med en søk/navigasjonsapplikasjon er presentert i kapittel 3 og vil videre behandles i utviklingsløpet med prototypen slik at vi prioriterer denne delen til andre forretningsmessige krav.



5.2.1 Søkemotoroptimalisering (SEO)

Som det fremkommer i avsnitt 3.5.4 er digital synlighet hos søkemotorene et strategisk mål. Dette vil innebære at Forbrukertipset.no må ha et teknisk og kvalitetsmessig nivå som er tilfredsstillende.

Vi går ikke inn på de kvalitetsmessige aspektene her siden det er behandlet tidligere, men vi vil likevel understreke viktigheten av at forretningssiden ser grundig på dette da brudd på søkemotorers retningslinjer faktisk kan gjøre nettstedet helt usynlig. Det er også en spesiell fare for dette med vår forretningsmodell (Affiliate) fordi autogenerated innhold uten relevant tekst satt i en kontekst, kan bli tolket som juks(Google, 2008).

5.2.1.1 Teknisk kvalitet

Vi vet vi må ha en tilfredsstillende teknisk kvalitet på nettstedet. Det vil underminere den kvalitetsmessige og den betalte innsatsen dersom dette ikke oppfylles, men hva er tilfredsstillende? Vi mener en av de beste kildene til å svare på dette spørsmålet er Google fordi de er markedsledende på søk både i Norge og på verdensbasis. I tillegg har dette selskapet gjort allment tilgjengelig en rekke publikasjoner som skal hjelpe nettsteder å bli synlige uten å bryte reglene til søkemotorene i samme prosess. Vi gjengir her en forkortet versjon av Google's tekniske retningslinjer som er relevant for oss(Google, 2008):

1. Bruke en tekstbasert nettleser som Lynx til å undersøke nettstedet, fordi de fleste søkemotorer ser siden på samme måte som denne. Dersom utstrakt bruk av JavaScript er tatt i bruk kan Lynx avsløre i hvor stor grad søkerboter har problemer med å gjennomse nettstedet.
 2. Sørg for at webserver støtter if-Modified-Since http header. Dette tillater webserveren å fortelle Google hvorvidt innholdet er endret siden forrige søkerobotbesøk. Støtte for dette sparer nettstedet for unødning bruk av båndbredde.
 3. Bruke robots.txt til å forhindre gjennom søking av søkeresultatsider eller annet autogenerated innhold som ikke tilfører særlig grad av verdi for brukere som kommer fra søkemotorer.
- I tillegg til retningslinjene ovenfor gjelder de faktorer som påvirker rankingen hos søkemotorer generelt. Igjen vil vi her understreke at søkemotoroptimalisering i denne sammenhengen er fra et teknisk perspektiv.



5.2.1.2 Andre momenter

Det er et mål fra forretningsområdet at applikasjonen fungerer i alle moderne nettlesere. Sagt på en annen måte: Nettstedet må være tilgjengelig for flest mulig brukere. Dette innebærer også Internet Explorer 6 og 7 siden de er markedsleder med ca 80% av markedet, men det er tekniske utfordringer med Internet Explorer som stiller høyere krav til utvikling enn andre moderne nettlesere som Firefox, Safari og Opera for å nevne de mest kjente. Vi mener at Yahoo sin gradering av nettlesere kan være nyttig i denne sammenheng for å avgjøre hva som kan og/eller bør støttes utover Internet Explorer. Yahoo graderer nettlesere i A, C og X. Hvor A er det som regnes for å være moderne nettlesere, det vil si de har støtte for de fleste funksjoner som eksisterer på moderne nettsteder. Det gir også en indikasjon på hvorvidt det Open Source baserte JavaScript biblioteket til Yahoo støtter nettleseren. C graderte nettlesere er ikke så vanlig lenger, dette er litt eldre nettlesere og er i realiteten i ferd med å fases ut. Yahoo støtter ikke disse. X graderte nettlesere er svært sjeldent, har ingen eller svært få funksjoner for å surfe på nett. Figuren 16 viser graderingstabellen til Yahoo (Yahoo, 2008).

	Win 98	Win 2000	Win XP	Win Vista	Mac 10.4	Mac 10.5
IE 7.0			A-grade	A-grade		
IE 6.0	A-grade	A-grade	A-grade			
Firefox 2.+	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade
Opera 9.+	A-grade	A-grade	A-grade		A-grade	A-grade
Safari 3.0+					A-grade	A-grade

Figur 16 – Yahoo's graderingstabell for nettlesere.

Vi mener Yahoo med dette har satt en akseptabel standard for hva som er hensiktsmessig å støtte av ulike nettlesere. Yahoo sier i samme artikkel at dette kombinert med testing mot Lynx nettleseren nevnt tidligere i dette kapittelet vil dekke ca 99% av alle Yahoo sine brukere. Dette skulle indikere at potensielle Forbrukertipset.no brukere også er ivaretatt.



5.3 Hvordan løse identifiserte krav fra Brukerområdet?

Fra Brukerområdets arbeid, som har pågått parallelt med utvikling av prototypen, har vi fått resultater fra undersøkelser som er gjort mot testpersoner. Resultatene fra disse undersøkelsene skaper et behov for endring av brukergrensesnittet. Samtidig har en stor del av brukerområdets arbeid vært å avdekke rent teoretisk hvilke faktorer og hensyn vi må ta ved utvikling av nettstedet slik at siden er tilgjengelig og informasjonen lett kan finnes. Resultatene fra disse undersøkelsene skaper et behov for endring av teknisk oppbygging av HTML struktur og CSS.

5.3.1 Grensesnitt og konsekvenser

Testresultatene fra brukerområdet forteller at det er noen punkter som alle var enige om. Disse resultatene gir krav til endring av grensesnittet og er å anse som svært viktige å forbedre i en endelig versjon. Disse endringene får konsekvenser utover de rent plasseringsmessige fordi applikasjonen er utviklet med utstrakt bruk av DOM. Dette vil innebære at dersom det blir gjort noe annerledes i html strukturen så må DOM skriptingen oppdateres fordi den bruker referanser til spesifikke html-tagger.

5.3.2 Kvalitet og tilgjengelighet

Foreløpig har ikke brukertesting avslørt et umiddelbart behov for endringer av teknisk kvalitet på nettstedet som retter seg mot tilgjengelighet. Våre brukertester har vært rettet mot applikasjonens logiske flyt og ikke om den er egnet for svaksynte, fargeblinde eller andre typer brukere som trenger å kontrollere elementer på siden. En endelig versjon av applikasjonen bør derfor utvikles med kvalitet i alle ledd. Dan Zederholm poengterer (Bulletproof Web Design, 2008) at bruk av pixler ikke er en fleksibel og god løsning for tekst fordi Internet Explorer 6 ikke lar brukeren forstørre tekst definert i pixler. Teknisk sett er årsaken at nettleseren ikke kan overstyre spesifikasjonene til designeren når det brukes pixler. I IE7 er det mulig for brukeren å forstørre, men denne zoomingen forstørrer hele skjermbildet og ikke bare teksten. I vår applikasjon vil det bryte med visjonen om alt hele navigasjonsprosessen skal foregå i samme skjermbilde fordi brukeren da blir nødt til å scrolle vinduet. Det finnes flere løsninger til denne type problematikk, vi kan for eksempel bruke CSS nøkkelord som definerer teksten til "small" og ut i fra den sette ulike elementers tekst til en viss prosent av denne. Det finnes flere løsninger som også løser dette på en effektiv måte og det vil øke nettstedets tilgjengelighet betraktelig.

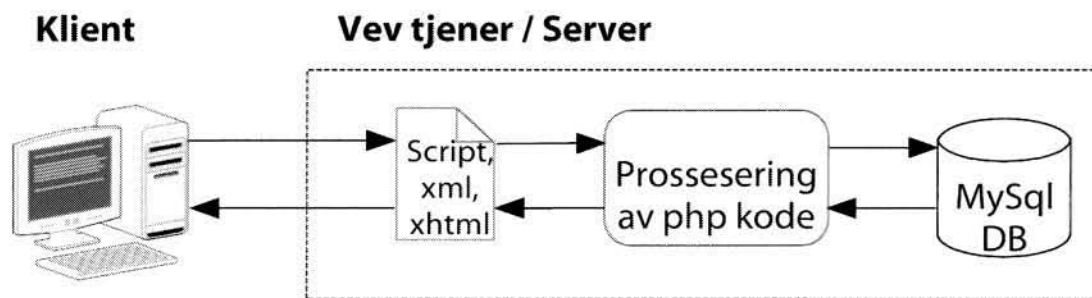
5.4 Realisering av visjonen (prototyp utvikling)

Redegjørelsen av valgt metode for utvikling av denne prototypen er omhandlet i kapittel 2 og vil således ikke berøres her. På grunn av den klare todelingen av utviklingsløpet vil de to delene, klient og tjener, bli behandlet separat i forbindelse med den videre redegjørelsen.

5.4.1 Tjenersiden – Utviklingsløpet

For å håndtere databehandlingen på serveren var det tre primærbehov som måtte dekkes. En server plattform, en database plattform og et kommandospråk(skript).

Figur 17 viser tjenerhåndtering for å motta og besvare forespørsler fra klient.



Figur 17 – Prinsippkisse for samspillet mellom klient og tjener.

En enkel HTML versjon av Forbrukertipset.no var allerede etablert på serveren til webhotellet PROISP¹⁰. Serverfunksjonaliteten til PROISP ble ansett for å være tilstrekkelig for å utvikle prototypen, samt drifte en mer fullverdig applikasjonen uten all for stor trafikk.

Når det gjaldt valg av databaseplattform falt valget på den sql baserte MySQL plattformen. Denne er integrert i webhotellet, var kjent for utvikler fra før og har fri kildekode. Siden det er xml data som skal behandles ble det vurder bruk av en "native XML database" plattform. En slik løsning ville gitt muligheter for å lagre xml dataene uten den prosesseringen som en sql basert løsning krever. En "native XML database" plattform kan forventes å gi gevinster ved spørring mot databasen kontra en sql database.

Da et slik veivalg ville krevd mye ressurser i form av innlæring av et nytt database spørrespråk, samt etablering av en ny databaseplattform på webserver og en utvikling av egendefinerte skriptbibliotek for oppkobling og spørringer ble denne plattformen inntil videre forkastet.

Dette vil være mer aktuelt senere når større datamengder skal behandles, samt at denne typen databaseplattformen blir integrert på webservere og i skriptspråkene.

¹⁰PROISP et linux basert webhotell med tilgang til PHP versjon 5 og MySQL versjon 5.



Når det gjelder skriptspråk er det mange og velge i og alle har sine sterke og svake sider. Vårt valg ble PHP uten å vurdere andre så nøye. Skriptspråket er modent og leveres som åpen kildekode. Det er regnet for å være meget godt egnet for utvikling av dynamiske og sikre webbaserte applikasjoner. Skriptspråket har en meget god integrering mot MySQL og fra versjon 5 også en godt og bredt spekter med metoder for xml håndtering. I tillegg er PHP som eneste skriptspråk innarbeidet hos utvikler og en del grunnfunksjonalitet var derfor allerede utviklet.

5.4.1.1. Plan for-/ og utvikling av serverfunksjonalitet

For å kunne håndtere forespørsler fra klienten på et senere tidspunkt måtte det på plass en databaseløsning på serveren som inneholdt flere tabeller med knyttinger til hverandre. Det måtte også på plass skript som kunne sortere ut ønskede produktdata fra xml filen fra Tradedoubler, samt legge disse på rett plass i databasen med de nødvendige knyttinger. For å kunne sortere og tilordne informasjonen tilgjengelig i xml filen var det også nødvendig å tilordne egendefinert tilleggsinformasjon.

5.4.1.1.1 Databaseløsning

Tilnærming til de nødvendige databasetabellene ble organisert rundt den allerede etablerte kolonnestrukturen i klientapplikasjonen, samt et ønske om å unngå lagring av samme data mange ganger.

Første kolonne i klientapplikasjonen skal vise hvilke interesseområder produktene er tilordnet. Det ble derfor opprettet en egen tabell for dette, "pr_mainc".

Neste kolonne i applikasjonen skal vise kategoriene som interesseområdet videre var brutt opp i. En tabell for dette, "pr_subca", ble opprettet med knytning til interesse id.

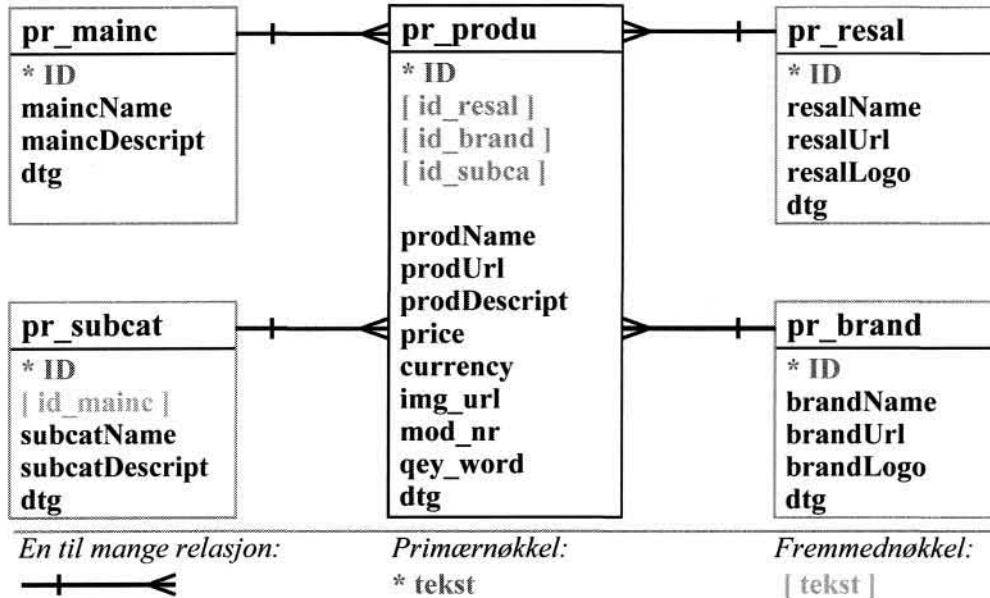
Neste kolonne skal vise alle produktene som er tilordnet aktuelle kategori og til slutt i fjerde og siste kolonne skal det spesifikke produktet presenteres.

Disse to kolonnene mente vi at med fordel kunne hente data fra samme databasetabell, "pr_produ". Denne tabellen måtte ha knytning til kategori id for å gi et håndterlig datagrunnlag.

Vi fant det videre formålstjenelig å skille ut informasjon om merke og forhandler i egne tabeller, "pr_brand" og "pr_resal". Dette for å unngå mye dobbeltlagring i databasen.

Det måtte derfor også opprettes koblinger til disse to tabellene i "pr_produ" tabellen.

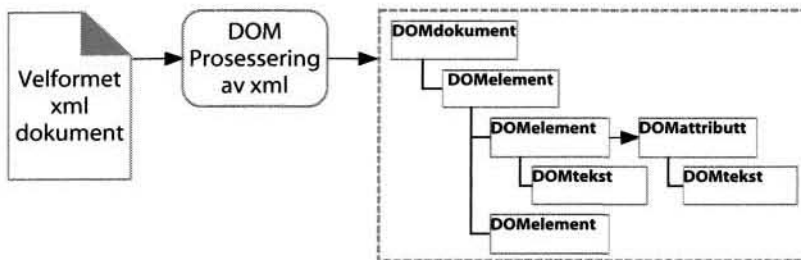
Det fullstendige bildet av tabellen og hvordan de henger sammen kan ses i figur 18.



Figur 18 – ER diagram som viser identifiserte behov for databasetabeller, samt deres knytninger.

5.4.1.1.3. Hvordan lese og sette sammen xml dokumenter

En viktig brikke for å realisere en egen database med produktdata er prosessering av xml dokument/filer. prosessering av xml filer sett i et tradisjonelt perspektiv, består i å bryte opp xml strukturen i fast definerte deler og tilordne disse delene i en trestruktur i dataminnnet.



Figur 19 – Prinsipp modell for DOM prosessering av xml.

Fra og med versjon 5 av PHP er xml håndtering en godt implementert funksjonalitet. De to mest benyttede funksjonene for dette er DOM funksjoner og SimpleXML funksjoner.

Selv om DOM funksjonen er vanskeligere å ta i bruk enn SimpleXML funksjonen ble denne funksjonaliteten førstevalget.

Grunnen til det ligger i at SimpleXML funksjonen har en noe begrenset funksjonalitet som vi antok ville legge unødige hindringer i veien for applikasjonsutviklingen senere. Det ble gjort vellykkede øvelser både med tanke på å lese og skrive xml dokumenter ved hjelp av DOM prosessering på små og relativt enkle xml dokument, men...

DOM funksjonene har en stor svakhet. Svakheten består i at hele xml dokumentet må leses inn i maskinens minne før data kan hentes ut. Da vi har behov for å lese og hente ut data av store xml filer vil denne metoden egne seg dårlig og sannsynligvis ikke mulig på et webhotell. En videre undersøkelse avslørte at php har innbygget funksjonalitet for xml prosessering ved hjelp av å streame dataene inn og ut av dataminnnet. Dette betyr at lesing av et xml dokument starter fra toppen, en fast definert mengde data tas inn i minnet og behandles, så fjernes disse dataene og en ny mengde tas inn helt til dokumentet er gjennomgått.

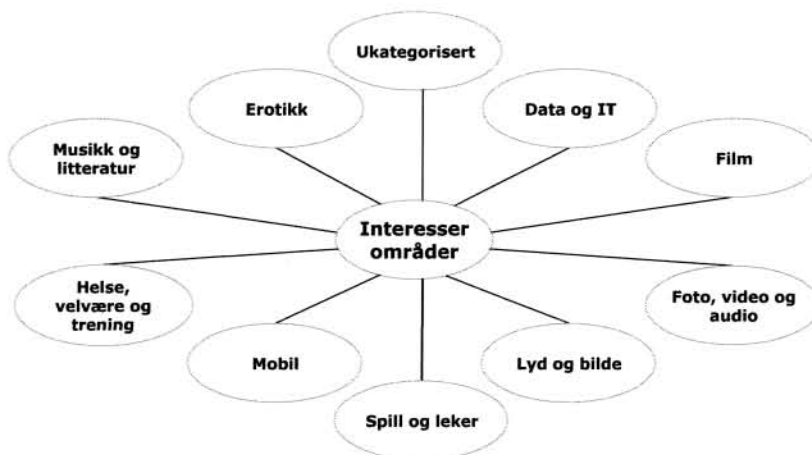
Valget av prosessering for lesing av xml dokumenter i php ble derfor endret til XMLReader funksjonen. En lignende streamer metode i php, Simple API for XML (SAX), ble også vurdert, men valget falt på XMLReader funksjonen da denne er regnet for å være lettere å ta i bruk og ha bedre funksjonalitet og ytelse.

Når det gjelder de xml dokumentene som skal settes sammen på serveren og sendes over til klienten, så er disse så små at DOM prosessering er å anse som en meget god løsning både med tanke på prototypen men også sett opp mot en mer fullverdig applikasjon.

Faktisk så er disse xml dokumentene av en så enkel struktur at bruk av SimpleXML funksjoner hadde vært fullt ut tilstrekkelig til prototypen. Men siden mer komplekse behov må forventes å melde seg senere velger vi å gå for DOM prosessering med en gang.

5.4.1.1.4. Hvordan fremskaffe egen tilleggsdata

Produkt dataene som vi laster ned i form av xml fil fra Tradedoubler inneholder allerede kategori data satt av Tradedoubler og i tillegg egne kategoridata satt av produktforhandlerne på noen produkter. Disse har vært utgangspunktet for tilordningen av våre egne kategorier. Men før dette ble vi enige om en interessefordeling som kunne fange opp produktspektret. De ni(ti) interesseområder måtte dekkes i forhold til produktspektret kan ses i figur 20:



Figur 20 – Interesseområdene til første kolonne i vår applikasjon.



Videre ble produktdataene for alle produktene fra Tradedoubler, ca 185.000 produkter, lagt over i et Microsoft Excel regneark. Det ble tatt med data om; Produktnavn, Tradedoublers produktkategori og forhandlers produktkategori. Disse ble videre sortert ved hjelp av innebygd funksjonalitet i Excel og denne sorteringen ga 492 unike sammensetninger. Videre måtte hver sammensetning manuelt tildeles et av de interesseområdene vi hadde opprettet og en egen hovedkategori som passet inn under interesseområdet, samt passet til produktsammensetningen, måtte også opprettes og tilordnes.

Samme Excel regneark ble brukt til å generere data til et `array` som var nødvendig i skriptet som skulle opprette databasetabellene `"pr_mainc"` og `"pr_subca"` med data. Dette array'et inngår i php skriptet, `fillMainSubCatDb.php`. Excel regnearket ble også brukt til å generere data til et annet array som kunne identifisere bindingen mellom egne hovedkategorier og hvert enkelt produkt tilgjengelig fra Tradedoubler.

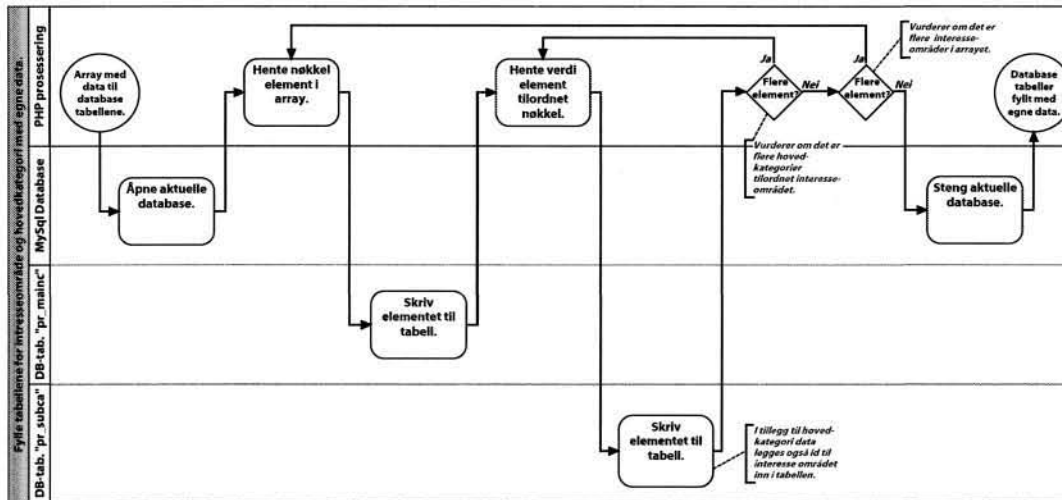
Array'et består av en Md5 hash verdi som nøkkel, denne er basert på hver unike kombinasjon av Tradedoubler's kategoribenevnelse pluss forhandler kategori benevnelse. Md5 verdiene ble opprettet i php skriptet `dataToLinkTab.php` som omtales senere. Videre er verdien som er tilordnet nøkkelen i array'er vår egen definerte hovedkategori. Dette array'et er brukt i php skriptet `addCatToLinkTab.php` som også omtales senere.

5.4.1.1.5. Legge ønskede data inn i databasetabellene

Tabellene "pr_mainc" og "pr_subca"

Gjennom manuelle grep i Excel filen hadde vi nå nok data til å opprette databasetabellene `"pr_mainc"` og `"pr_subca"`. Det ble derfor opprettet et php skript, `fillMainSubCatDb.php`, som kunne fylle begge databasetabellene med egne data og knytte de sammen. Datagrunnlaget for skriptet er array'et opprettet med egendefinerte data fra Excel regnearket.

Figur 21 viser gangen i php skriptet.

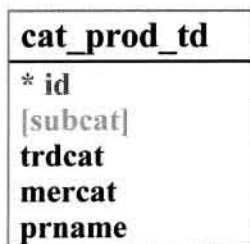


Figur 21 – Dataflyt modell for skript: fillMainSubCatDb.php

Videre er kildekoden til php skriptet fillMainSubCatDb.php gjort tilgjengelig i vedlegg 7.

Tilleggstabellen "cat_prod_td"

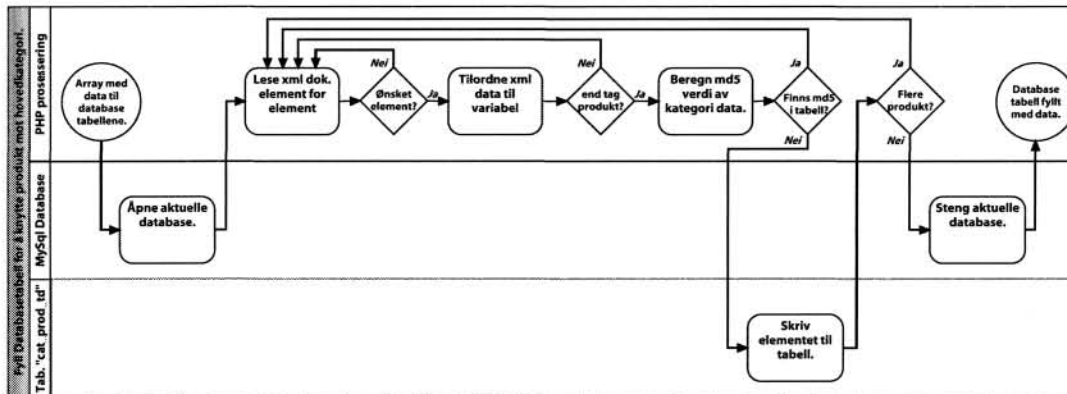
Videre ble det identifisert et behov for å opprette en ekstra databasetabell som kunne knytte hovedkategoriene og produktene sammen. Dette kunne gjøres ved at hver unike kombinasjon av Tradedoubler og forhandlerkategori ble kodet om til en md5 hash streng og denne ble knyttet opp mot vår egen hovedkategori. Databasetabell som vist i figur 22 ble derfor opprettet.



Figur 22 - ER diagram for databasetabell "cat_prod_td"

Id i denne tabellen vil utgjøre md5 verdien, subcat vil tilsvare id for hovedkategorien som id (md5 verdien) skal knyttes mot. Trdcat, mercat og prname er videre bare identifiseringsverdier for id (md5 verdien) og har ingen praktisk betydning. Det ble videre laget et eget php skript for å generere data til den nye databasetabellen. Php skriptet, dataToLinkTab.php, leser xml produkt dataene fra Tradedoubler, element for element. For hvert produkt genereres en md5 streng basert på Tradedoublers kategoribenevnelse og forhandlers kategori benevnelse.

Strengen skal utgjøre en unik id i databasetabellen så før ny rad blir opprettet sjekkes det om denne id er opprettet tidligere i tabellen. Hvis ikke blir ny rad opprettet i tabellen og id tilordnes genererte md5 verdi. Subcat blir satt til tom, dette kommer vi tilbake til, mens trdcat, mercat og pname tilordnes tilhørende verdier lest inn fra xml filen. Figur 23 viser gangen i php skriptet.



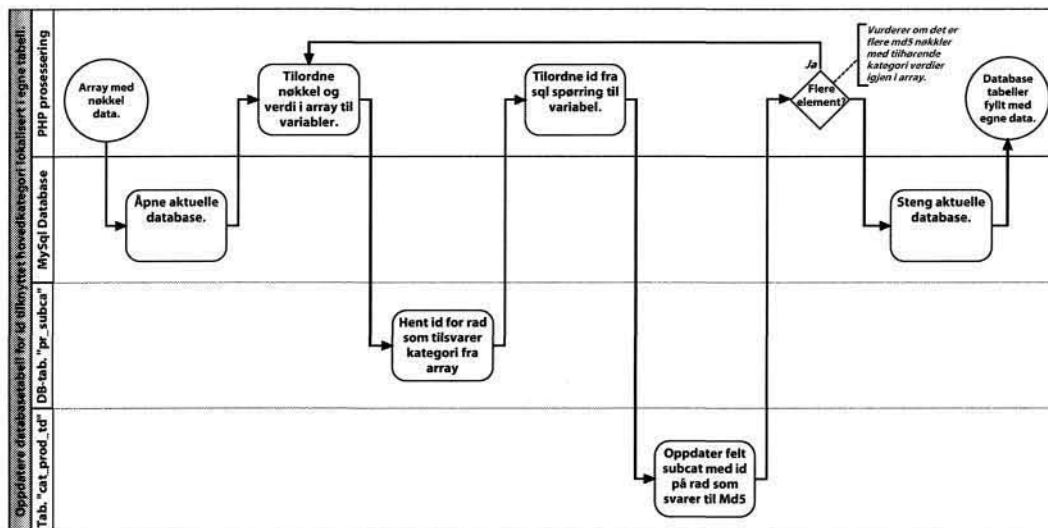
Figur 23 – Dataflyt modell for skript: dataToLinkTab.php

Videre er kildekoden til php skriptet dataToLinkTab.php gjort tilgjengelig i vedlegg 8.

Da var databasetabellen for å knytte produkt og hovedkategori på plass med unntak av at subcat feltet som står tomt.

For å fylle dette feltet ble det opprettet et eget php skript, addCatToLinkTab.php. Skriptet er bygget rundt array'et som knytter md5 verdiene til en hovedkategori som ble laget utfra Excel regnearket.

I skriptet sjekkes så hver database rad mot array'et og rette verdi for subcat blir tilordnet. Figur 24 viser gangen i php skriptet.



Figur 24 – Dataflyt modell for skript: addCatToLinkTab.php

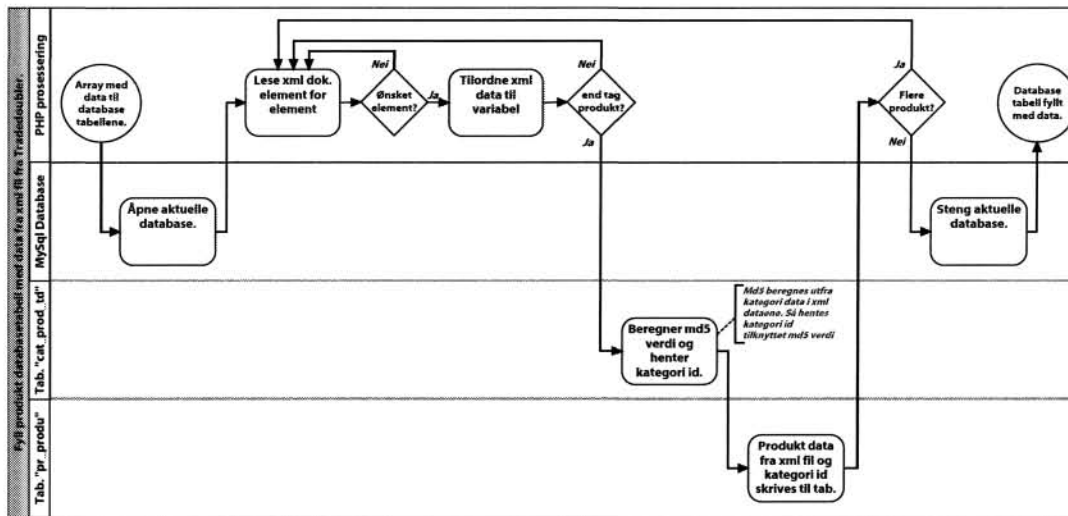
Videre er kildekoden til php skriptet addCatToLinkTab.php gjort tilgjengelig i vedlegg 9.

Tabellen "pr_produ"

Med grunndataene på plass var det nå på tide å opprette hovedtabellen "pr_produ" som skal inneholde de nødvendige produktdataene, samt nødvendige knyttinger til hovedkategori tabellen "pr_subca".

Det var i utgangspunktet tatt høyde for implementering av egne tabeller med informasjon og knytning til forhandler og merke: "pr_brand" og "pr_resal". Dette viste seg å være vanskelig å få på plass og ble derfor forkastet inntil videre da det ikke utgjør noen kritisk funksjonalitet opp mot prototypen. Skriptet, readToProdTab.php, leser xml dokumentet, element for element, ved bruk av XMLReader funksjonen. De dataene som skal inn i databasen tilordnes egne variabler. Når XMLReader treffer på end taggen for et produkt, </product>, Skal innsamlede data skrives til databasetabellen. I tillegg beregnes md5 verdien basert på Tradedoublers kategoribenevnelse og forhandlers kategori benevnelse. Med denne verdien som utgangspunkt hentes id til hovedkategori fra databasetabellen "cat_prod_td". Denne id gir den viktige bindingen mellom hovedkategori og produkt.

Da kan dataene skrives til databasetabellen og neste produkt kan leses fra xml filen. Figur 25 viser gangen i php skriptet.



Figur 25 – Dataflyt modell for skript: readToProdTab.php

Videre er kildekode til php skriptet readToProdTab.php gjort tilgjengelig i vedlegg 10.

5.4.1.2. Behandle forespørsler fra klienten

Før de nødvendige databasetabellene var opprettet og på plass med data, hadde utvikler av klientsiden behov for å teste sin applikasjon med reelle forespørsel til tjener.

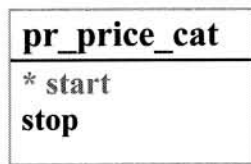
Det ble derfor etablert en felles forståelse av hvilke data som skulle leveres til de fire forskjellige kolonnene i klientapplikasjonen og et midlertidig skript ble etablert på tjenersiden for å få dette til å fungere. Med denne midlertidige datainfrastrukturen på plass var derfor grunnfunksjonalitet allerede opparbeidet. Det som måtte tilpasses var databasespørringene som måtte gjøres for å skaffe data til hver kolonne, samt DOM funksjonaliteten som skulle sette sammen xml dokumentet til klienten.

Når vi så kunne kjøre klientapplikasjonen mot hele produktbasen vår, ca 180.000 produkter, erfarte vi at fire kolonner ikke ga nok filtreringsmuligheter. Valg av hovedkategori i kolonne to ga veldig mange produkter til kolonne tre. For å bøte på dette i prototypen avgjorde vi at vi skulle innføre en ekstra kolonne mellom kolonne to og tre.

I denne kolonnen valgte vi å bruke pris som en ekstra filtreringsmulighet.



Det ble derfor utarbeidet en egen databasetabell, som vist i figur 26.



Figur 26 - ER diagram for databasetabell "pr_price_cat"

Start utgjør minimumsbeløp og er samtidig tabellens unike id, mens stop angir maksimumsbeløpet. Følgende min/max beløp ble lagt inn i databasetabellen.

- start: 1, 100,200,300,500,750,1000,2000,3000,4000,6000,8000,10000,12000,14000,18000,22000,30000
- stop : 100,200,300,500,750,1000,2000,3000,4000,6000,8000,10000,12000,14000,18000,22000,30000,99000

I tillegg måtte det implementeres ekstra funksjonalitet i skriptet readToProdTab.php, skriptet som skriver produktdata til produkt databasetabellen. Den ekstra funksjonaliteten går ut på at det legges inn en referanse til priskategori når produktdata skrives til databasetabellen på samme måte som det legges inn en referanse til hovedkategori. Priskategorien tilordnes basert på prisen på produktet.

Denne funksjonaliteten inngår i koden som er gjort tilgjengelig i vedlegg 10, men har ikke tidligere vært omtalt.

Serverløsningen som skal respondere på forespørsler fra klienten er delt inn i fire separate skript. I tillegg brukes generiske databasefunksjoner tilordnet i egen skriptfil. Koden for denne er gjort tilgjengelig i vedlegg 11. De fire applikasjonsavhengige skriptene er: **xmlOutput**, kildekode er gjort tilgjengelig i vedlegg 12.

Skriptet som klienten kaller. Behandler input fra klienten og med bakgrunn i input kalles rette funksjon for videre behandling og respons.

script_init, kildekode er gjort tilgjengelig i vedlegg 13.

Setter opp strukturen for serverapplikasjonen med hvilke mapper og skript som skal inngå. Det er først og fremst databasehåndteringsfunksjoner som kalles her.

genXml, kildekode er gjort tilgjengelig i vedlegg 14.

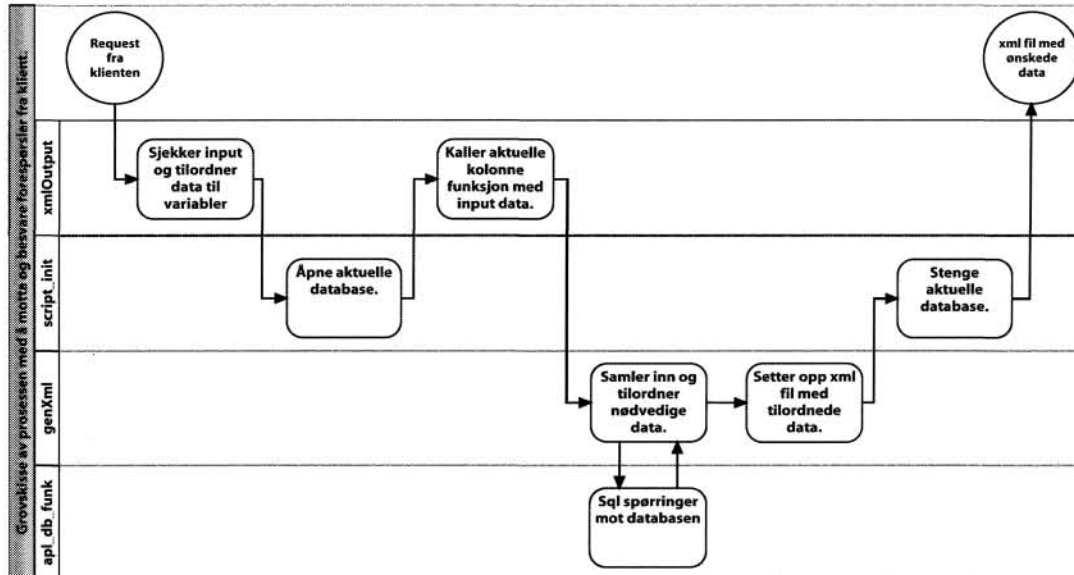
Skriptet er delt inn i flere funksjoner, en funksjon for hver kolonne som skal fylles med data på klienten. I disse funksjonene genereres data ut fra input, disse tilordnes et xml dokument med en struktur som klienten kjenner til og gjøre seg nytte av.



`apl_db_funk`, kildekode er gjort tilgjengelig i vedlegg 14

Skript som håndterer alle sql spørringene mot databasen.

Figur 27 viser samspillet mellom php skriptene for å besvare en forespørsel fra klienten.



Figur 27 - Grovskisse av prosessen med å motta og besvare forespørsler fra klient.

5.4.1.2.1. Hvilke input må serveren ha for å respondere korrekt.

Når det gjelder første kolonne trenger serverskriptet ingen input parametere. Bare ved å kalle skriptet, `xmlOutput.php`, iverksettes funksjonen for generering av data til kolonne `en` og følgende xml data sendes over til klienten.

```
<?xml version="1.0"?>
<menu>
  <level>1</level>
  <cat id="2">Data og IT</cat>
  <cat id="4">Film, musikk og litteratur</cat>
  <cat id="5">Foto, video og audio</cat>
  <cat id="6">Helse, velvære og trening</cat>
  <cat id="7">Hjem og bolig</cat>
  <cat id="8">Lyd og bilde</cat>
  <cat id="9">Mobil</cat>
  <cat id="10">Spill og leker</cat>
  <alt>8</alt>
</menu>
```

Teksten til element `<level>` forteller hvilken kolonne det gjelder. Teksten til element `<cat>` sier hva interesseområdet er, mens tilknyttet attributtverdi til `id` sier hvilken id interesseområdet har i databasen. Teksten til element `<alt>` sier hvor mange `<cat>` elementer som er levert.



For å respondere med korrekt data til kolonne **to** på for eksempel valg av ”mobil” fra kolonne **en** trenger serveren følgende input: col=2, id=10. Dette forteller server skriptet at det skal leveres data til kolonne **to** (col=2) og at hovedkategorier tilknyttet id 10 under interesseområder skal sendes tilbake. Følgende xml data sendes over til klienten.

```
<?xml version="1.0"?>
<menu>
  <level>2</level>
  <cat id="75">Diverse spill og leker</cat>
  <cat id="76">Brettspill</cat>
  <cat id="77">Dataspill</cat>
  <cat id="78">Leker</cat>
  <cat id="79">Spillmaskiner</cat>
  <paging page="1"/>
  <alt>5</alt>
</menu>
```

Her har elementene samme tilordning og betydning som ved forrige kolonne. Men i tillegg er elementet <paging> med. Dette er for å imøtekomme en fremtidig funksjonalitet og har ingen betydning for prototypen.

For å respondere med korrekt data til kolonne **tre** på for eksempel valg av ”brettspill” fra kolonne **to** trenger serveren følgende input: col=3, id=76. Dette forteller serverskriptet at det skal leveres data til kolonne **tre** (col=3) og at produkter tilknyttet id 76 under hovedkategorier skal sendes tilbake. Følgende xml data sendes over til klienten.

```
<?xml version="1.0"?>
<menu>
  <level>3</level>
  <cat id="100">200</cat>
  <cat id="200">300</cat>
  <cat id="300">500</cat>
  <cat id="500">750</cat>
  <alt>76</alt>
</menu>
```

Dette utgjør en filtrering av tilgjengelige brettspill på pris. Attributtet id angir minimumsprisen mens element teksten angir maksimumsprisen. I tillegg er hovedkategori id fra forrige kolonne sendt med som tekst til elementet <alt>.

For å respondere med korrekt data til kolonne **fire** på for eksempel valg av prisfiltrering 300-500 fra kolonne **tre** trenger serveren følgende input: col=4, id=300, alt=76. Dette forteller serverskriptet at det skal leveres data til kolonne **fire** (col=4) og at produkter tilknyttet id 76 under hovedkategorier med en pris mellom kr 300 og kr 500 skal sendes tilbake.



Følgende xml data sendes over til klienten.

```
<?xml version="1.0"?>
<menu>
  <level>4</level>
  <cat id="58463">Cranium</cat>
  <cat id="58465">Monopol</cat>
  <cat id="58467">Trivial Pursuit Familie</cat>
  <cat id="58468">Trivial Pursuit Disney Norge</cat>
  <cat id="58469">Trivial Pursuit Geni 2006 Edition</cat>
  <cat id="58470">RISK</cat>
  <cat id="58477">Kvitt eller Dobbelt</cat>
  <cat id="58488">Pro Poker Aluminium Poker sett</cat>
  <paging page="1"/>
  <alt>8</alt>
</menu>
```

Elementene har samme betydning som under kolonne **tre**, men tilknyttet attributtverdi til id forteller nå hvilken produkt id i databasen det gjelder.

For å respondere med korrekt data til kolonne **fem** på for eksempel valg "RISK" fra kolonne **fire** trenger serveren følgende input: col=5, id=58470. Dette forteller server skriptet at det skal leveres data til kolonne **fem** (col=5) og at produkt med id lik 58470 skal det levers data på. Følgende xml data sendes over til klienten.

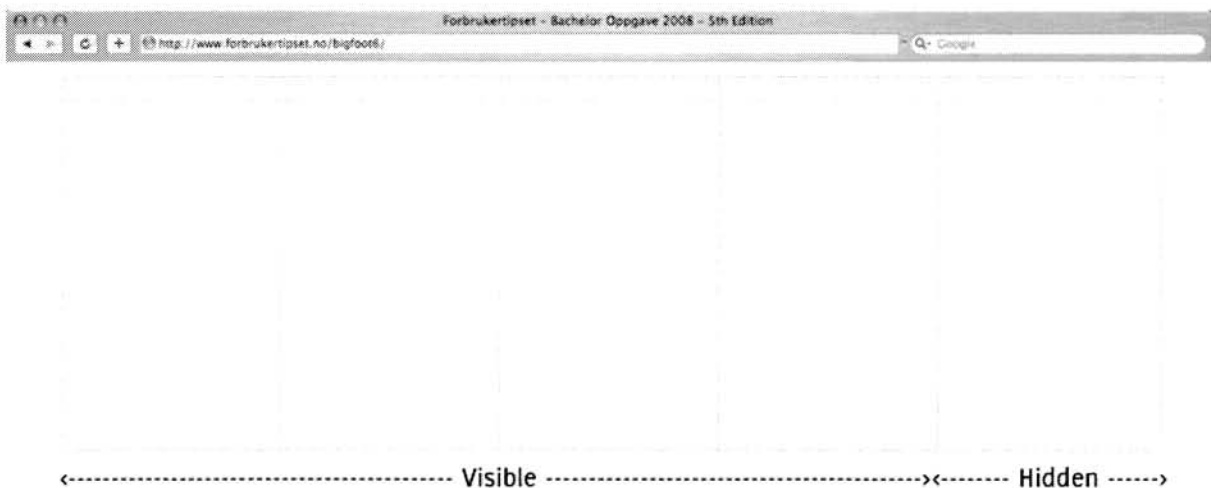
```
<?xml version="1.0"?>
<product>
  <level>5</level>
  <merchant>1</merchant>
  <name>RISK</name>
  <description>Fra 10 år, 2-6 spillere</description>
  <image_url>http://www.mpx.no/img/produktbilder/Brettspill.....</image_url>
  <buy_url>http://pdt.tradedoubler.com/click?a(284570)p.....</buy_url>
  <price>395.00</price>
  <search_name>MPX</search_name>
  <category>tom</category>
  <brand>tom</brand>
  <alt>1</alt>
</product>
```

Teksten til element <merchant> angir pr nu den standard id som skal knytte produkt til en forhandler. Som tidligere redegjort for er databasetabell for forhandler og merke ikke på plass. Teksten til element <name> viser navnet på produktet. Teksten til element <description> viser informasjon rundt produktet. Teksten til element <image_urt> gir en link til et produktbilde. Teksten til element <buy_urt> gir en link til websiden hvor produktet kan kjøpes. Teksten til element <price> viser prisen på produktet. Teksten til element <search_name> viser produkt forhandler og dette er noe som inn til videre er tatt inn under qey_word i produkt databasen. Teksten til element <category> og <brand> viser tom fordi disse ikke er i bruk i dag. Teksten til element <alt> sier hvor mange produkter som er levert, altså 1.

5.4.2. Klientsiden – Utviklingsløpet

Vår ide baserer seg på at brukeren kan finne sitt ønskede produkt ved å navigere fra et overordnet interesseområde og ned til produktet i samme skjermbilde. Trinnene i denne prosessen skjer ved at brukeren filtrerer for eksempel i kategorier og pris underveis. Det som skal gjøre applikasjonen unik og attraktiv er at denne prosessen foregår uten at siden lastes på nytt for hvert valg brukeren tar av filtreringsvalg, og at alt skjer i samme skjermbilde, uten behov for scrolling. Samtidig skal informasjonen lastes dynamisk etter hvert som brukeren ber om den, og fjernes automatisk dersom brukeren går vekk fra den aktive kategorien han eller hun befinner seg i.

Dette gir oss tre primærbehov på klientsiden. Det er et grensesnitt som ivaretar kravene om samme skjermbilde i hele prosessen, det er et opplegg for asynkron lasting av data basert på brukerens valg, og det er muligheten for å vise eller gjemme informasjon basert på hvilken kolonne brukeren befinner seg i. Figuren 28 viser hvordan applikasjonen er tenkt.



Figur 28 – Kolonnefordeling i prototypen

5.4.2.1 Grensesnittet

Vi har valgt å bruke XHTML (Extensible Hypertext Markup Language) og CSS (Cascading Style Sheet) til strukturering av grensesnittet. XHTML er valgt fordi det stiller strengere krav til korrekt koding enn vanlig HTML.

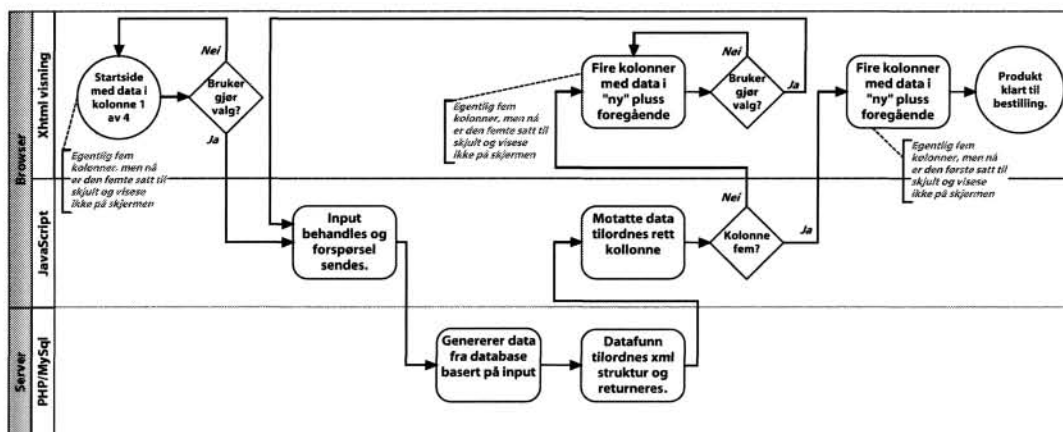
Vi mener det er bedre å forholde seg til dette rent generelt, men til prototyping spesielt da utstrakt bruk av DOM manipulering og Ajax vil benyttes sammen med XML som datakilde.

Det gir også flere fordeler som totalt sett er grunnlaget for denne beslutningen.

<http://www.w3.org/MarkUp/2004/xhtml-faq#advantages>

CSS er valgt fordi DOM og Ajax krever referanser til hvilke elementer som skal opprettes, endres eller slettes. En separering av styingen til disse elementene gjør det betydelig enklere i tillegg til at dette er regnet for å være et prinsipp for god kodeutvikling.

Prototypen skal vise fire horisontale kolonner i skjermbildet. Disse skal vises i sin helhet uten behov for scrolling, dog er det ikke tatt hensyn til skjermopløsning under 720px vertikalt. Et viktig krav til applikasjonen er at informasjonen ikke lastes før brukeren foretar et valg. Dette gjelder alle kolonner bortsett fra første kolonne som laster alle interesseområder ved initiering. Som nevnt tidligere i denne rapporten oppsto det et behov for en femte kolonne og det er derfor behov for en løsning slik at denne ikke vises før det er behov, og at første kolonne da gjemmes slik at det til enhver tid kun vises fire kolonner i skjermbildet. Figur 29 illustrerer dette.



Figur 29 – Flyttdiagram over prosessen med å fylle kolonnene med data.

(X)HTML strukturen for denne designløsningen er forholdsvis enkel. Vi har opprettet alle kolonnene som div elementer i en overordnet tabell. All stying er tatt hånd om av CSS filen og vi bruker JavaScript til å manipulere DOM, slik som å vise eller gjemme kolonne fem.

Alle kolonnene er kodet på samme måte og figuren 30 viser (X)HTML strukturen for første kolonne.



```

<td class="level" id="level-1">
<h3 class="series" id="series-1"></h3>
<div class="pane">
<div class="browse-total" id="total-1"></div>
<div class="paginate" id="navlinks1"></div>

<ul id="one">
<!-- Her fylles interesseområder -->
</ul>

</div>
</td>

```

```

<!--Kolonne-->
<!--Kolonnenavn-->
<!--Datamottager-->
<!--Infoholder-->
<!--Pagingholder-->

```

Figur 30 - (X)HTML strukturen for første kolonne

Som det fremkommer av figur 30 har vi en (X)HTML struktur med id og/eller klassereferanser, men uten verdier. I nettleseren gir dette oss som vi har sett i avsnitt 5.4.2 en tabell med fire tomme kolonner. All informasjon blir dermed lastet inn dynamisk etter hvert som navigasjonsprosessen forløper. Dette gjelder både selve XML dataen brukeren etterspør og annen informasjon som kolonnenavn, totale resultater og paging. Vi har også tatt hensyn til behov for informasjon som ikke er med i prototypen. Vi har to div- tagger til disposisjon som er tiltenkt å gi brukeren informasjon om det totale antallet som finnes i kolonnen, og en som skal gi brukeren en mulighet til å page dersom det er flere resultater enn kolonnen har plass til i høyden.

Figuren 31 viser stylingen til kolonnestrukturen ovenfor.

```

.level {
  vertical-align: top;
  width: 227px;}

#browse table {
  border: 1px solid #b3b3b3;
  border-collapse: collapse;}

#browse table td {border: 1px solid #b3b3b3;
  overflow: hidden;
  width: 226px; }

#browse .pane {
  padding: 1px;
  font-size:98%;
  margin-bottom: 14px;
  min-height: 350px;}

```

Figur 31 - Styling til kolonnestrukturen

Når det gjelder prototypen har vi valgt å se bort i fra alt som ikke er av vesentlig betydning i forhold til viktige spørsmål som må besvares og unik funksjonalitet som må bevises gjennomførbart. Vi mener derfor dette er tilstrekkelig for å gå videre til spørringer mot server og håndtering av respons.

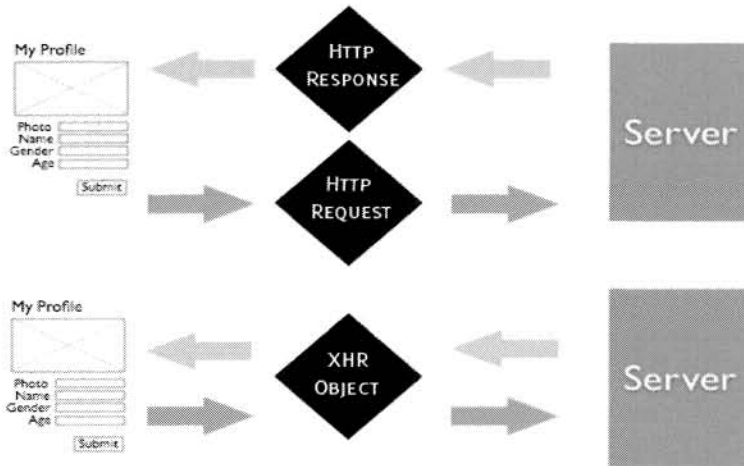
5.4.2.2 Dynamisk lasting av data



For å kunne laste data dynamisk og asynkront inn etter hvert som brukeren ber om informasjon tar vi i bruk Ajax. Dette er en teknikk som benytter flere teknologier og opprinnelig er dette en forkortelse for Asynchronous JavaScript og XML, men det er senere blitt vanlig å benevne dette som et navn i stedet for en forkortelse siden teknikken kan bruke andre kilder enn bare XML.

Det er uansett en nyttig teknikk som vil gi oss den funksjonaliteten vi er ute etter. Samtidig er det forbundet med kostnad som vi er nødt til å ta med i hele regnestykket. For det første vil dette innebære at JavaScript må være aktivert i brukerens nettleser. Dersom det ikke er det vil applikasjonen ikke fungere, men problemet kan likevel løses ved å tilby en løsning for de brukerne som faller utenfor. Dette kan være en forenklet versjon av applikasjonen som mangler vesentlige attributter, men som likevel fungerer med grunnfunksjonaliteten inntakt. I denne sammenheng brukes det begreper som ”progressive enhancement” og ”graceful degradation” som henholdsvis viser til å berike nettstedet med JavaScript og la nettlesere som mangler funksjonalitet likevel få fortsette, selv om det ikke blir så elegant (Advanced DOM Scripting, 2007). Dette innebærer med andre ord mer enn bare Ajax, dette vil gjelde all bruk av JavaScript. Sett i forhold til denne prototypen har vi ikke tatt hensyn til dette da vi anser det for å være utenfor de viktigste spørsmålene. Det andre problemet med Ajax er knyttet til de ulike metodene nettleserne håndterer dette objektet. Det vil si hvordan Internet Explorer 5 og 6 løser dette, og hvordan resten løser det. Vi mener det ikke er nødvendig med en historie om verken Ajax eller hvordan de ulike nettleserne håndterer dette her siden Internet Explorer 7 og 8 bruker samme metode som resten og fordi utfordringene mer ligger i DOM skriptingen enn selve XHR (XmlHttpRequest) objektet. Det er mer den økte responsfølelsen og de mulighetene som Ajax gir vi ønsker å demonstrere her.

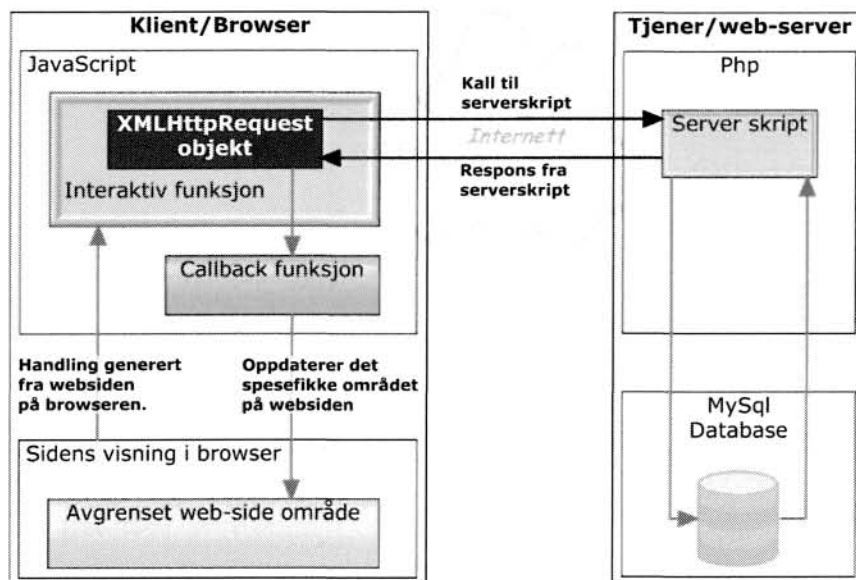
Figuren 32 viser hvordan Ajax fungerer annerledes enn tradisjonell klient/tjener spørringer.



Figur 32 - Ajax versus tradisjonell klient/tjener spørringer

I motsetning til tradisjonelle applikasjoner hvor brukerens valg sendes til serversiden som en http spørring som deretter kommer tilbake til klientsiden og oppdateres ved at nettleseren oppdaterer hele vinduet så fungerer XHR objektet annerledes. Her håndteres både spørring og respons av XHR objektet og nettleseren oppdaterer ikke hele vinduet. I stedet oppdateres bare det området som er påvirket av spørringen.

Det hele foregår på en slik måte at responsfølelsen blir betydelig sterkere og det gir også nye muligheter til hvordan vi kan designe bedre løsninger for brukerne. Vi har utviklet Prototypen med Ajax, sentralt i Ajax teknikken ligger dette XHR objektet. Figuren 33 viser i mer detalj hvordan objektet sender og mottar data.



Figur 33 - XHR objektets rolle

Vi har valgt å ta i bruk et XHR objekt som er såkalt `cross browser` (håndterer forskjeller mellom de ulike nettleserne). Likevel, det er ikke et mål med denne prototypen at den fungerer i alle nettlesere og vi anser ikke det for å være en del av de sentrale spørsmålene vi ønsker svar på med prototypen. Prototypen er som nevnt utviklet med fokus på våre definerte problemstillinger og dette innebærer at andre utfordringer som er viktige å løse i en endelig versjon er utelatt. Likevel har vi forsøkt å holde oss til standarder og beste praksis der det ikke har påvirket oss tidsmessig i nevneverdig grad.

Selve XHR objektet er et eksempel på kode som fungerer i alle moderne nettlesere, men som ikke har påvirket oss tidsmessig. Figuren 34 viser koden.

```
function sendRequest(url, callback, postData) {
    var req = createXMLHTTPObject();
    if (!req) return;
    var method = (postData) ? "POST" : "GET";
    req.open(method, url, true);
    req.setRequestHeader('User-Agent', 'XMLHTTP/1.0');
    if (postData)
        req.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
    req.onreadystatechange = function () {
        if (req.readyState != 4) return;
        if (req.status != 200 && req.status != 304) {
            return;
        }
        callback(req);
    }
    if (req.readyState == 4) return;
    req.send(postData);
}

var XMLHttpRequestFactories = [
    function () {return new XMLHttpRequest();},
    function () {return new ActiveXObject("Msxml2.XMLHTTP");},
    function () {return new ActiveXObject("Msxml3.XMLHTTP");},
    function () {return new ActiveXObject("Microsoft.XMLHTTP");}
];

function createXMLHTTPObject() {
    var xmlhttp = false;
    for (var i=0; i<XMLHttpRequestFactories.length; i++) {
        try {
            xmlhttp = XMLHttpRequestFactories[i]();
        }
        catch (e) {
            continue;
        }
        break;
    }
    return xmlhttp;
}
```

Figur 34 – Kode for XHR objektet

5.4.2.3 DOM Manipulasjon

Manipulering av DOM (Document Object Modell) vil være en betydelig del av kodearbeidet på klientsiden. Det meste av dette er rettet mot HTML DOM, men også XML DOM blir brukt. Vi har et behov for å sende en forespørsel om informasjon til tjener, motta responsen og fylle første kolonne med denne informasjon.

Samtidig oppretter vi kolonnenavn og andre støtteindikatorer fortløpende med DOM. Vi starter denne prosessen ved å be om de elementer vi ønsker fra det XML dokumentet som vi har mottatt fra tjeneren.

Vi pakker denne informasjonen inn i de (X)HTML elementene vi har definert med id og/eller klasse og brukeren vil dermed ha første kolonne og de respektive interesseområdene klare.

Figuren 35 viser hvordan dette presenteres.

Interessefelt	
Data og IT	▶
Erotikk	▶
Film, musikk og litteratur	▶
Foto, video og audio	▶
Helse, velvære og trening	▶
Hjem og bolig	▶
Lyd og bilde	▶
Mobil	▶
Spill og leker	▶

**Velg
Interesse**

Figur 35 - Presentasjon av første kolonne.

Fra dette punktet vil kolonne to fylles med hovedkategorier og hvilke hovedkategorier dette er, baseres på hvilket interesseområde brukeren velger. Samtidig som at kolonne to fylles med hovedkategorier vil hjelpeindikatoren (Velg Interesse) måtte fjernes og kolonnenavn for kolonne to må legges til. Vi bruker samme prosedyre som for kolonne en når det gjelder henting og håndtering av tjenerresponsen. Figuren 36 viser neste trinn i navigasjonsprosessen.

Interessefelt	Hovedkategorier	
Data og IT	Diverse hjem og bolig	▶
Erotikk	Husdyr	▶
Film, musikk og litteratur	Kikkert	▶
Foto, video og audio	Kjøkkenapparater	▶
Helse, velvære og trening	Kjøleskap	▶
Hjem og bolig	Kontaktlinser	▶
Lyd og bilde	Mikrobølgeovn	▶
Mobil	Oppvarming	▶
Spill og leker	Støvsugere	▶
	Vaskemaskiner	▶

**Velg
hovedkategori**

Figur 36 - Neste trinn i navigasjonsprosessen.

Det er samme DOM håndtering vi benytter i hele denne navigasjonsprosessen. Samtidig dukker det opp nye utfordringer som må løses med mer DOM manipulering. Utfordringen er at alle kolonner må tømmes som ikke er den aktive kolonnen brukeren står i, eller den påfølgende som viser hvilke elementer som befinner seg under det aktive valget.

Samtidig må alle hjelpeindikatorer nullstilles og opprettes på nytt slik at det reflekterer brukerens posisjon. Figuren 37 viser kildekoden for første kolonne, men denne gangen slik den blir generert i DOM.

```
<td class="level" id="level-1">                                <!--Kolonne-->
<h3 class="series" id="series-1"></h3>                       <!--Kolonnenavn-->
<div class="pane">                                           <!--Datamottager-->
<div class="browse-total" id="total-1"></div>                <!--Infoholder-->
<div class="paginate" id="navlinks1"></div>                 <!--Pagingholder-->

<ul id="one">
<li><a onclick="selectCat(this, '2', '2', '0')" title="Velg Data og IT" href="#" class="coll">Data og IT</a></li>
<li><a onclick="selectCat(this, '2', '4', '0')" title="Velg Film, musikk og litteratur" href="#" class="coll">Film, musikk og
litteratur</a></li>
<li><a onclick="selectCat(this, '2', '5', '0')" title="Velg Foto, video og audio" href="#" class="hi">Foto, video og audio</a></li>
<li><a onclick="selectCat(this, '2', '6', '0')" title="Velg Helse, velvære og trening" href="#" class="coll">Helse, velvære og trening</
a></li>
<li><a onclick="selectCat(this, '2', '7', '0')" title="Velg Hjem og bolig" href="#" class="coll">Hjem og bolig</a></li>
<li><a onclick="selectCat(this, '2', '8', '0')" title="Velg Lyd og bilde" href="#" class="coll">Lyd og bilde</a></li>
<li><a onclick="selectCat(this, '2', '9', '0')" title="Velg Mobil" href="#" class="coll">Mobil</a></li>
<li><a onclick="selectCat(this, '2', '10', '0')" title="Velg Spill og leker" href="#" class="coll">Spill og leker</a></li>
</ul>

</div>
</td>
```

Figur 37 - Kildekoden første kolonne med DOM generering.

5.4.2.4 Oppsummering

Fra forretningsområdet var den første og mest sentrale oppgaven å utvikle en prototyp webapplikasjon som kunne bekrefte eller avkrefte om ideen var gjennomførbar. Det skulle brukes feeds i XML format fra Tradedoubler, disse skulle kontrolleres på egen server i egen database. Denne informasjon skulle så gjøres tilgjengelig for brukere av Forbrukertipset.no hvor de skulle kunne søke/navigere i informasjonen uten at nettleteren oppdaterte innholdet for hvert steg. Hele prosessen skulle foregå i samme skjermbilde. Har teknologiområdet klart å realisere dette?

Vi har gjennom arbeidet med prototypen bevist at vi behersker håndteringen av feeds fra Tradedoubler og inn til egen database. Ikke bare har vi tatt informasjonen umodifisert, men vi har organisert disse og lagt til egne data slik at vi kan ha maksimal kontroll. Videre har vi bevist at bruk av Ajax som teknikk lar seg gjennomføre slik at brukere kan navigere i informasjonen slik det er beskrevet. Samtidig er det flere momenter som kompliserer og øker vanskelighetsgraden slik at det må gjøres mer arbeid for at dette kan settes ut i livet som en forretningsapplikasjon.



På serversiden gjelder det i første omgang håndtering av feeds fra andre leverandører enn Tradedoubler. Det er relativt store forskjeller på hvordan disse er satt opp slik at et opplegg for håndtering og sammenfletting av disse er ønskelig. På klientsiden krever det mer arbeid for å løse grensesnittproblemer slik at applikasjonen virker i de moderne nettleserne. I dag virker prototypen ikke i noen av Internet Explorer versjonene.

Den gjør det i de andre moderne nettleserne slik at målsettingen bare delvis er oppnådd. Fra forretningsområdet er det også viktige strategiske momenter som reiser flere utfordringer. Å gjøre denne applikasjonen søkemotorvennlig vil kreve ytterligere undersøkelser og utvikling. Det er også en reell mulighet for at slik utstrakt bruk av JavaScript ikke lar seg forene med forretningsidens ambisjoner om søkemotorrangering. Fra brukerområdet avslører tester at strukturering av elementer på siden er feilplassert og derfor ikke blir sett. Dette er i utgangspunktet uproblematisk utfordringer, men på grunn av krav til hvordan applikasjonen skal fremstå visuelt og prosessmessig vanskeliggjør dette utviklingen. Sett i lys av dette kan vi oppsummert si at vi har klart imøtekomme de viktigste og fleste forutsetningene til applikasjonen, men at det samtidig gjenstår flere viktige utfordringer. Når det gjelder en konklusjon på hovedproblemstilling og underproblemstillinger vil vi behandle dette i kapittel 6 sammen med de andre strategiområdene slik at vi får en samlet behandling av problemstillinger.



6. Konklusjoner vedrørende problemstillingene

6.1. Konklusjoner vedrørende forretningsområdet

For å kunne møte de krav en realisering av vårt prosjekt har de forretningsmessige aspekt forbundet med dette hatt som hovedformål å sette føringer for hvordan prosjektet skal kunne nå sine satte mål. Samspillet mellom forretningsområdet, brukerområdet og teknologiområdet har stått sentralt i dette.

6.1.1 Forretningsområdets overordnede problemstilling

Hvilke fortrinn og markedsstrategier kan bidra til lønnsomhet?

Vi har i hele prosjektperioden hatt lønnsomhet som et nødvendig og forventet resultat som må komme før definerte tidsgrense nås. Gruppen mener det er for tidlig å si hvorvidt våre fortrinn og markedsstrategier faktisk bidrar til lønnsomhet samlet sett.

6.1.2 Forretningsområdets første underproblemstilling.

Hvilke fortrinn må vi fokusere på i forhold til konkurrentene?

Visjonen vår er å utvikle en applikasjonen som håndterer søk/navigasjon på en annerledes måte enn dagens tradisjonelle søkemotorer og informasjonsstruktur. Samtidig mener vi vår løsning er bedre for brukeren fordi informasjonen presenteres mer oversiktlig og applikasjonen er svært enkel og intuitiv å bruke. Dette er hva vi kaller innovative løsninger som er klare fortrinn i forhold til våre konkurrenter og som vi derfor må fokusere spesielt på.

6.1.3 Forretningsområdets andre underproblemstilling.

Hvilke kostnadseffektive markedsføringstiltak vil passe vår situasjon?

Vi konkluderer med at det er påkrevet med en spesiell fokus på søkemotoroptimalisering og tilrettelegging i forhold til optimalisering av markedsføringstiltak. Dette fordi digital synlighet er den mest kostnadseffektive markedsføringen vi kan gjøre i dag. Dette er derfor å anse som nye strategiske mål som også kan berøre andre utviklingsområder. Vedrørende den betalte markedsføringen konkluderer vi med at overvåking av hva som kommer tilbake for hver anvendte krone (ROI) og fortløpende korrigeringer av annonser for å være beste løsning. Den betalte markedsføringen kombinert med søkemotoroptimalisering er hva vi anser for å passe vår situasjon.



6.1.4 Forretningsområdets tredje underproblemstilling

Hvordan definere lønnsomhet: Innsats, tid og fortjeneste?

Innsatsen vår er dem tiden vi bruker til å utvikle konseptet målt i penger. Dette er igjen tildelt en tidsramme mot gitte milepæler. Fortjenesten inn mot milepæl blir således en generert inntekt basert på trafikk og kjøpe via vår internett side som utlikner innsatsen. En nullpunktsanalyse vil således gi svar på om dette er tilfellet.

De øvelsene som er simulert gjennom bruk av modeller for beregning av return of investment, sett i markedsføringssammenheng har gitt videre input til en estimert kontantstrøm for prosjektet. Det faktum at lønnsomhetsanalysen kom ut med store forskjeller ut fra hvilken konverteringsrate som ble benyttet forteller oss at det er en god del usikkerhet forbundet med en slik etablering. Derfor må lønnsomhet forsøkes oppnådd over en gitt tidsperiode, dette er den eneste måten å finne det sikkert ut på.

6.1.5 Forretningsområdet oppsummert

Vi har gjennom valgte problemstillinger skissert opp hva som gruppen anser for å være viktig for å lykkes med dette nettstedet. Forretningsområdet har identifisert en forretningsmodell som skal være gjeldene i oppstartsfasen, videre er det identifisert konkrete fortrinn med vår applikasjon i forhold til konkurrenter, og det er identifisert budsjettammer for innstas, tid og fortjeneste. Totalt sett anser vi forretningsideen til å være gjennomførbar.

6.2. Konklusjoner vedrørende brukerområdet

For å komme brukeren i møte i en tidlig utviklingsfase har vi lagt vekt på gjennomføring av brukertester, samt en tilstrebing av beste praksis beskrevet i relevant teori inn mot utforming av vårt grensesnitt.

Prototypen vi har utviklet skal ikke bare være teknisk i stand til å oppfylle visjonen definert under foretningsområdet, men skal også være utformet slik at brukeren fort og intuitivt oppfatter hva prototypen kan brukes til og videre hjelpe brukeren til raskt å finne det vedkommende er på jakt etter. Dette er ikke bare en viktig brukerstrategi for teknisk utvikling, men også svært viktig inn opp mot forretningsstrategi for et rent nettbasert forretningskonsept.

6.2.1. Brukerområdets overordnede problemstilling

Oppfyller prototypen forretningsvisjonen sett fra et brukerperspektiv?

Vi mener, selv ut fra et spinkelt test grunnlag, at forretningsvisjonens punkter om en løsning som er intuitivt lett å forstå og har et brukervennlig grensesnitt er godt løst i prototypen.



Vi mener også uavhengig av brukertestene at prototypen har en innovativ tilnærming til produktsorteringstjenester på internett. Dette betyr ikke at prototypen er perfekt sett i et brukerperspektiv, men dette omtales senere.

6.2.2. Brukerområdets første underproblemstilling

Skjønner brukerne vårt innovative produktfiltrerings konsept?

Brukertestene har vist oss at kolonnestrukturen for å presentere valgbare elementer er enkel å forstå og kan med rett utforming gi en rask og oversiklig måte å navigere på mellom hundretusenvís av produkter. I tillegg skal denne utformingen bidra til å gjøre oss unike i forhold til konkurrentens mer tradisjonelle tilnærming. Det er derfor svært viktig for oss å kunne slå fast at dette ser ut til å være en effektiv, brukervennlig, spennende og innovativ måte å gjøre produktsortering på som faktisk viser seg å fungere mot reelle brukere.

Videre vil identifiserte endringsbehov være et viktig input og ha med seg videre i utviklingen av applikasjonen. I all hovedsak kan endringsbehov fra testene kokes ned til at den statiske fordelingen av jevnstore kolonner som vi startet med viste seg å være en feil tilnærming. Kolonnen må kunne tilpasses i størrelse i forhold til det informasjonsbehovet de skal fylle. Behovet for bredere kolonner for å kunne presentere mer informasjon for brukeren gjør at andre kolonner igjen må skjules. Dette gir et behov navigasjon tilbake til skjulte kolonner. Dette var forsøkt løst i prototypen med en klikkbar stolpe til venstre i applikasjonen. Denne var det ingen av testpersonene som fikk øye på den. Denne navigasjonsmuligheten må derfor gjøres langt mer synlig i videreutviklingen av applikasjonen.

6.2.3. Brukerområdets andre underproblemstilling

Følger utviklet nettapplikasjon etablerte brukerkonvensjoner?

Vi har identifisert brudd meg brukerkonvensjoner som må rettes opp, men i tillegg må det sies at vår prototyp ikke faller inn under standard oppsett heller og dermed er ikke alle standard konvensjoner nyttige for oss å vurdere. Følgende konvensjonsbrudd ønsker vi derimot å ta hensyn til i videre utvikling av nettapplikasjonen.

Relevant teorien har bekreftet funn fra testene om at søkeboksen var vanskelig å få øye på da den ikke er plassert i henhold til gjeldende konvensjoner. Konvensjonen tilsier plassering i nettsidens hode. Relevant teori brakte også en løsning til et identifisert behov for mer informasjon om hva applikasjonen er for noe i tillegg til intensjoner om å tilføre flere tjenesteapplikasjoner på nettstedet i nær fremtid.



Løsning går ut på og etablere en egen startside som kun skal informere brukeren om hva som finnes på nettstedet, samt introdusere en overordnet navigasjon mellom nettsidene. Dette vil være viktige faktorer å ta med seg i den videre utviklingen av et nettsted med en helhetlig profil og enkel navigasjon.

6.2.4. Brukerområdet oppsummert

Vi kan derfor klart konkludere med at prototypen oppfyller visjonen brukermessig. Videre har tester og relevant teori avdekket forbedringspotensial og utviklingsretninger som er viktige å ta med seg i den videre utviklingen av nettstedet.

6.3. Konklusjoner vedrørende teknologiområdet

Våre problemstillinger og metoder for problemløsning innen teknologistrategipunktet har gitt oss viktige svar på hva som kreves for en realisering av Forbrukertipset.no. Den eksperimentelle utviklingen av en fungerende prototype har i denne sammenheng vært en sentral del av våre metoder og vi har gjennom dette tilegnet oss kunnskaper som i stor grad preger våre konklusjoner. I tillegg til dette har krav og føringer både fra forretningsstrategipunktet og brukerstrategipunktet påvirket hva som er sentralt.

6.3.1 Teknologiområdets overordnede problemstilling

Hvilke teknologiske og tekniske forutsetninger vil ligge til grunn for en realisering av applikasjonen?

Gjennom den eksperimentelle utviklingen av prototypeapplikasjonen har gruppen kunne ta i bruk kunnskaper om utvikling fra tidligere, men vi har også måttet tilegne oss ny kompetanse. Fra tidligere hadde vi noe erfaring på tjenersiden med PHP og MySQL, men i denne sammenheng måtte vi lære oss å håndtere store datamengder i form av XML i tillegg. På klientsiden hadde vi fra før erfaring med (X)HTML og CSS, men i denne sammenheng måtte vi ta i bruk Ajax og DOM. I utviklingsprosessen har vi derfor fått frisket opp i ferdigheter gruppen var i besittelse av før prosjektstart, og vi har også tilegnet oss nye ferdigheter som applikasjon krevde. Samlet sett har vi derfor et godt bilde av hva som kreves av utviklerkompetanse for et slikt nettsted. Vi har en fungerende applikasjon i dag og vi besitter derfor den viktigste kompetansen som er nødvendig i denne sammenheng. Samtidig har vi foretatt prioriteringer underveis og det er derfor en rekke mindre problemstillinger som må håndteres i den endelige versjonen.



Disse er ikke utfordringer som krever spesiell kompetanse, de er heller ikke en del av de forutsetningene som ligger til grunn, men mer av kvalitetsmessig karakter.

6.3.2 Teknologiområdets første underproblemstilling

Hvordan løse andre strategiske momenter fra forretningsstrategipunktet?

Et prosjekt av denne typen med flere ukjente faktorer har krevd nødvendig korrigerende underveis i utviklingsløpet. Vi visste ikke i startfasen hva som kunne tenkes å dukke opp av nødvendige momenter. Her har digital synlighet kommet frem som en viktig strategisk målsetting fra forretningsområdet som har direkte innvirkning på valg av teknologi og teknisk gjennomføring. Digital synlighet krever en kombinasjon av innholdsmessig kvalitet og teknisk kvalitet. For teknologiområdet vil det være tilstrekkelig å overholde de tekniske retningslinjene til Google og disse innebærer ikke vesentlige endringer av applikasjon eller kode. I tillegg til digital synlighet er det prioritert at nettstedet er tilgjengelig på alle moderne nettlesere. Dette er en ensidig positiv konsekvens for teknologiområdet fordi dette innebærer en naturlig heving av kvalitet uten endring i teknologi eller teknisk gjennomføring.

6.3.3 Teknologiområdets andre underproblemstilling

Hvordan løse identifiserte krav fra brukerstrategipunktet?

Endringsbehovet fra brukerområdet er i det store og hele av en slik art at det blir en del av den pågående utviklingen og vedlikehold. Først og fremst er det endringer i informasjonsstruktur som kreves og dette løses fortløpende. Det er i dag ingen av disse kravene som krever endring i teknologi eller teknisk gjennomføring.

6.3.4 Teknologiområdet oppsummert

Begge underproblemstillingene slik de fremstår i dag blir håndtert på en tilfredsstillende måte med teknisk gjennomføring som løsningsmetode.

Vi konkluderer derfor med at gruppen har identifisert forutsetningene som kreves oppfylt, og vi innehar den utviklerkompetansen som kreves for realisering av nettstedet sett fra et teknologisk og teknisk perspektiv.



7. Referanser litteratur/ressurser

Trykte kilder

- Beard, Jason (2007). *The Principles of Beautiful Web Design*, SitePoint Pty. Ltd.
- Berg Hansen, T., Hjertø, G. (2003). *Kvalitet og programvareutvikling* (1.utg.), Gyldendal Norsk Forlag AS
- Cederholm, Dan (2008). *Bulletproof Web Design*, New Riders.
- Faawang, O., Hoff, K.,G., Røine, E. & Westhaugen, H. (2004) *Prosjektarbeid, Utviklings- og endringskompetanse*, Gyldendal Akademiske
- Fernstrøm, L., Krogh, V.,G., Roos, G. & Roos, J.. (2005). *Strategi: en innføring* (4.utg.). Bergen: Fagbokforlaget
- Furu, Nina (2006). *Webkommunikasjon*, IJ-forlaget.
- Ghauri, P. Grønhaug, K. (2002) " *Research Methods in Business Studies – a Practical Guide* ", Prectice Hall Europe, 2 edition
- Harmon, Paul. (2007). *Busieness prosess change*, Morgen Kaufmann Publisne
- Johannessen, A. Kristoffersen, L. Tufte, P. (2004). *Forskningsmetode for økonomisk-administrative fag*. Abstrakt forlag
- Loveday, Loveday & Niehaus, Sandra (2008) *Web design for ROI*, New Riders.
- Nielsen, Jacob (1989). *Usability engineering at a discount*, Elsevier Science Inc.
- Perarlson og Saunders (2006) *Managing & using informations systems*, A strategic approach, Wiely
- Sambells, Jeffrey (2007). *Advanced DOM Scripting, Dynamic Web Design Techniques*, Friendsof.
- Sydik, Jeremy J. (2007). *Design Accessible Web Sites*, The Pragmatic Bookshelf

Internett kilder

- Berg, Lisbet (2006). *Forbrukerkompetanse* Lastet 26/03.08
fra:http://www.sifo.no/files/file66667_prosjektnotat_2006-7_web.pdf
- Bråthen, Gunnar (2003). *Ebusiness-Grunnleggende forretningsmodeller*. Lastet 04/04.08
fra:http://www.handel.no/modules/module_111/news_item_view.asp?iResponse=3&iNewsId=1525&iCategoryId=154
- Cristensen Claton M. (2005). *A roadmap to disruptive growth*. Lastet 15/5.08



fra:http://harvardbusinessonline.hbsp.harvard.edu/b02/en/common/item_detail.jhtml;jsessionid=D2QJTBZCOWTDAAKRGWDSSELQBKE0YIISW?id=9580&referral=2665&requestid=321868

- EIAA (2008) *Online shoppers 2008*, Lastet 15/05.08
<http://www.eiaa.net/news/eiaa-articles-details.asp?id=158&lang=1>
- Google. (2008) *Retningslinjer for Webmastere*. Lastet 14/05.08
 fra:<http://www.google.com/support/webmasters/bin/answer.py?answer=66361>
- Google. (2008) *Retningslinjer for teknisk kvalitet ved søkemotoroptimalisering*. Lastet 14/05.08
 fra:<http://www.google.com/support/webmasters/bin/answer.py?answer=35769>
- Kjørstad, Ingrid & Slette-meås, Dag (2008). *Digital mestring – internett og bredbånd, prosjektnotat nr. 2*. Lastet 25/03.08
 fra:http://www.sifo.no/files/file72900_p2-2008web.pdf
- Lavik, Randi & Schøll, Alexander (2007). *Markedet for hvite/brunevarer i Norge, en forstudie, Prosjektnotat nr. 4* Lastet 24/03.08
 fra:http://www.sifo.no/files/file71769_prosjektnotat2007-4.pdf
- Rappa, Micahal (2008). *Business models on the web*. Lastet 04/04.08
 fra:<http://digitalenterprise.org/models/models.html>
- Slette-meås, Dag (2007). *Forbrukernes stilling i informasjonssamfunnet, oppdragsprosjekt nr. 15*. Lastet 26/03.08
 fra:http://www.sifo.no/files/file72356_oppdagsrapport2007-15web.pdf
- Statistisksentralbyrå. (2007) *Modell, Utvikling i netthandel*. Lastet 26/03.08
 fra:<http://www.ssb.no/ikthus/fig-2007-09-20-03.gif>
- W3.org. (2004) *Bruk av XHTML og fordeler*. Lastet 18/05.08
 fra:<http://www.w3.org/MarkUp/2004/xhtml-faq#advantages>
- Wikipedia (2008). *Affiliate konsept* Lastet 15/05.08
http://en.wikipedia.org/wiki/Affiliate_marketing
- Yahoo.(2008) *Gradering av nettlesere hos Yahoo*. Lastet 15/05.08
 fra:<http://developer.yahoo.com/yui/articles/gbs/>

Annet

- Forelesningsmateriale tilgjengelig på HIBO fronter, samt egne notater.



8. Appendiks - Rapportens vedlegg

VEDLEGG TIL BACHELOR RAPPORT

1. SAMARBEIDSFORMER	2
2. MILEPÆLSPLAN FOR BACHELORPROSJEKTET:	3
3. ORDFORKLARINGSTABELL	4
4. OVERSIKT OVER GRUNNLEGGENDE FORRETNINGSMODELLER PÅ INTERNETT	8
5. INFORMASJON TIL INFORMANT I FORKANT AV BRUKERTEST	14
6. DOKUMENTSTRUKTUR PÅ XML DATA FRA TRADEDUBLER.	15
7. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: FILLMAINSUBCATDB.PHP	16
8. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: DATATOLINKTAB.PHP	17
9. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: ADDCATTOLINKTAB.PHP	18
10. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: READTOPRODTAB.PHP	20
11. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: FUNK_DATABASE.PHP	22
12. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: XMLOUTPUT.PHP	23
13. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: SCRIPT_INIT.PHP	24
14. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: GENXML.PHP	25
15. PHP KODE FOR SERVERSIDEN AV PROTOTYP. SKRIPT: APL_DB_FUNK.PHP	30
16. XHTML KODE FOR PROTOTYPEN. INDEX.HTML	31
17. CSS KILDEKODE FOR PROTOTYPE. HERO.CSS	33
18. KILDEKODE JAVASCRIPT FUNKSJONSBIBLIOTEK. SWEETLIB.JS	36
19. KILDEKODE JAVASCRIPT FIL. NEWENGINE.JS	37



Vedlegg til Bachelor rapport

1. Samarbeidsformer, verktøy og utfordringer.

Rapportens tre hovedfokusområder ble fordelt internt i gruppen slik at hvert prosjektmedlem hadde hovedansvaret for et av områdene. Videre har to av gruppens medlemmer hatt et delt ansvar for kodingen av prototypen. Det har også vært bidratt mot alle fokusområdene fra hele gruppen, spesielt med tanke på og identifiserer samspillet med fokusområdene.

Gruppen har kommunisert ved hjelp av møter, telefon, mail og msn. For å holde styr på oppgavegjennomføringen og samhandlingen har en nettbaserte prosjektstyrings applikasjon¹ vært benyttet. Rapporten er utarbeidet i Microsoft Word. Modeller er laget ved hjelp av tegnesoftwaren Smardraw og bildebehandlingssoftwaren Adobe Photoshop. Koding mot prototypen er gjort i editoren phpDesigner 2007. I tillegg har Microsoft Excel vært brukt til sortering og analyser av produktdataene fra Tradedoubler for å etablere egne tilleggsdata.

Prosjektet har hatt et veldig vidt perspektiv og jobbe etter. Det har derfor vært en stor utfordring å finne de gode konkrete problemstillingene som ivaretar fokusbredden, men også gir mulighet for spesialisering og fordypning. Vi synes vi har funnet denne balansen i oppgaven. Koding av prototypen ble også mer omtattende enn beregnet, dels på grunn av at det var vanskelig å beregne omfanget, men også fordi vi ønsket en godt grunnlag å kunne kjøre brukertester mot.

¹ Ace project (www.aceproject.com)



2. Milepælsplan for Bachelorprosjektet:

- 31 jan. Grovplan med problemstilling, disposisjon og undersøkelsesopplegg.
[BOPP krav]
- 10 feb. Ramme for prototyp utvikling klar.
- 10 mar. Prototyp applikasjon utviklet slik at tiltenkt sorteringsfunksjonalitet basert på egne database data kan demonstreres live.
- 31 mar. Muntlig presentasjon av prosjektet skal være gjort.
[BOPP krav].
- 09 apr. Fullverdig rapportstruktur utarbeidet med grunnlag i bearbejdede og konkrete problemstillinger.
- 14 apr. Plan for brukertesting utarbeidet.
- 28 apr. Fullverdig prototyp ferdig kodet iht. plan. Koden skal være strukturert og tilfredsstillende kommentert.
- 5 mai. Brukertester skal være gjennomført.
- 13 mai. Rapportens innhold er utarbeidet i en eller annen form.
- 21 mai. Rapport er klar for levering.
- 22 mai. Frist for innlevering av Bachelor oppgave.



3. Ordforklaringstabell

Affiliate

En affiliate er et nettsted som har innehar avtale(r) med en eller flere produkt / tjeneste leverandører. Ofte administreres interaksjonen mellom disse av partnernetverk

<http://en.wikipedia.org/wiki/affiliate>

Affiliate Network

Ett affiliate nettverk er et partnernetverk som har til hensikt å være bindeledd mellom leverandører og tilbydere. Affiliate nettverket gir leverandørene tilgang til et stort antall nettsteder som vil markedsføre produkter og tjenester. Nettstedene som vil markedsføre produkter får tilgang til et stort antall leverandører

http://en.wikipedia.org/wiki/Affiliate_networks

Ajax

Ajax, eller Asynkron JavaScript og XML, er en webutviklingsteknikk for å lage interaktive nettsider. Tanken er å lage nettsider som føles mer responsive. Dette gjøres ved at sidene utveksler litt og litt data med serveren i bakgrunnen, i stedet for å laste hele siden på nytt hver gang brukeren gjøre en forandring.

Ajax er ikke en teknologi i seg selv, men en betegnelse som refererer til en gruppe teknologier:

- HTML (eller XHTML) og stilark (CSS) for presentasjon.
- DOM (Document Object Model) manipuleres ved hjelp av JavaScript for å dynamisk generere informasjon.
- XMLHttpRequest-objektet for å kommunisere asynkront med en web-server.

[http://no.wikipedia.org/wiki/Ajax_\(Internett\)](http://no.wikipedia.org/wiki/Ajax_(Internett))

Array (php)

I php kan flere variabler med felles knytning tilordnes en gruppe(objekt) som kalles **array**.

Hver variabel i et array blir tilordnet en egen unik ID slik at dataene er lettere tilgjengelig.

Det opereres med tre forskjellige typer array's:

- Numerisk array Et array med numerisk ID (hvis ikke angitt så fra 0 og oppover).
- Assosiativt array Et array hvor hver ID er en brukergenerert tekstlig assosiasjon.
- Multidimensjonalt array Et array som inneholder flere array's (kan være en kombinasjon at de andre).

http://www.w3schools.com/php/php_arrays.asp

Cookie

Informasjonskapsel (engelsk HTTP cookie) lagres på platelageret for at nettleseren skal kunne ivareta personlig informasjon fra forskjellige vevtjenere på nettet, f.eks. autentisering og sporing av brukernavigering. Bruk av informasjonskapsler gjør det enkelt for programmerere å håndtere f.eks. brukerpreferanser og elektronisk handlekurv. Informasjonskapsler ble introdusert i 1994 av Netscape Communications Corporation i Netscape Navigator for å ta vare på tilstandsinformasjon, noe som ikke støttes i HTTP.

<http://no.wikipedia.org/wiki/Cookie>

Cross browser

Cross-browser refererer til evnen en nettside, vevapplikasjon, HTML struktur eller klientside script har til å støtte alle nettlesere. Betegnelsen cross-broser blir ofte blandet med multibrowser, men det er ikke riktig. Multi-browser betyr at noe virker på flere nettlesere, mens cross-browser betyr at det virker i alle nettlesere.

<http://en.wikipedia.org/wiki/Cross-browser>

CSS

CSS, eller Cascading Style Sheets er et språk som brukes til å definere utseende på filer skrevet i HTML eller XML. Prinsippet er at HTML- eller XML-dokumentet utelukkende skal beskrive struktur og semantikk, mens oppsett, farger og annen stilinformasjon skal beskrives ved hjelp av CSS. Stilinformasjonen kan integreres i selve dokumentet eller skilles ut som en egen fil med endelsen .css. Et ubegrenset antall dokumenter kan peke til og styres av samme .css-fil, noe som er styrken i CSS: Ved å endre på en fil, kan man endre fargebruk, bakgrunnsbilder osv. i alle dokumenter som peker til CSS-fil.

http://no.wikipedia.org/wiki/Cascading_Style_Sheets



CTR

Klikkrate er en måte å måle suksessen til en internett markedsføringskampanje. Klikkraten måles ved å dele antall klikk på en annonse på antall ganger en annonse har blitt vist (visninger). For eksempel, dersom en banner-annonse har blitt vist 100 ganger og 10 personer klikker på den, vil annonsen ha en klikkrate på 10%.

På engelsk er klikkrate referert til som CTR eller Click-through rate.

<http://no.wikipedia.org/wiki/Klikkrate>

DOM

Document Object Model (DOM) er en beskrivelse av hvordan en HTML- eller XML-trestruktur er representert. DOM tilbyr et objektorientert rammeverk for å analysere (parse) HTML eller XML til en veldefinert trestruktur og endre ets innhold.

http://no.wikipedia.org/wiki/Document_Object_Model

Elektronisk data interchange

Bedrifters applikasjon til applikasjon kommunikasjon for utveksling av data i standard formater for forretningstransaksjoner, ofte forkortet EDI

http://en.wikipedia.org/wiki/Electronic_Data_Interchange

Feed

Datafeed er en mekanisme for databrukere å motta oppdatert data fra en datakilde. Det blir vanligvis brukt i real-time applikasjoner, i point-to-point settinger og på web. På web kalles det også for web feed. Produkt feeds spiller en stadig større og viktigere rolle innen e-handel og internet marketing.

http://en.wikipedia.org/wiki/Data_feed

HTML

HyperText Markup Language (HTML, hypertextmarkeringsspråk) er et markeringsspråk for lagning av nettsider med hypertext og annen informasjon som kan vises i en nettleser. HTML benyttes til å strukturere informasjon – angi noe tekst som overskrifter, avsnitt, lister og så videre – og kan, i en viss grad, brukes til å beskrive utseende og semantikk i et dokument. HTMLs grammatiske struktur er HTML DTD som ble skapt ved å gjøre bruk av SGML syntaks, som er en internasjonal standard for tekstformatering (ISO 8879).

<http://no.wikipedia.org/wiki/HTML>

JavaScript

JavaScript er en implementasjon av ECMAScript, et skriptspråk som er best kjent for å tilføre dynamiske elementer til nettsider. Vanlige bruksområder er å bytte ut, fjerne eller legge inn tekst avhengig av hvor på siden du klikker, å skifte fokus i et skjema, og å åpne pop-up-vinduer.

<http://no.wikipedia.org/wiki/JavaScript>

Klient

En klient er en applikasjon som kobler seg opp mot en tjener eller annen datamaskin via et nettverk, og utfører handlinger deretter. En nettleser er et eksempel på en klient. Den kobler seg opp mot en HTTP- eller FTP-tjener og leser sidene, eller laster opp/ned filer.

[http://no.wikipedia.org/wiki/Klient_\(IKT\)](http://no.wikipedia.org/wiki/Klient_(IKT))

Konverteringsrate

Konverteringsrate er en måte å måle effekten av markedsføring på Internett[1]. Dette gjøres ved å måle forholdet mellom andelen av besøkende og en ønsket hendelse ved besøk av en internettside. En ønsket hendelse kan være salg, registrering, forespørsel eller nedlasting.

<http://no.wikipedia.org/wiki/Konverteringsrate>

Lynx

Lynx er en gratis open source, kun tekst, nettleser og Gopher klient for bruk på cursor-addressable terminaler.

http://en.wikipedia.org/wiki/Lynx_browser



MD5 (hash)

MD5 (Message-Digest algorithm 5) er en sjekksumalgoritme og en internettstandard (RFC 1321), som brukes av en mengde forskjellige sikkerhetsapplikasjoner og for å sjekke integriteten til datafiler. MD5-algoritmen lager en 128-bits (16 byte) sjekksum, som vanligvis oppgis som et 32-tegns heksadesimalt tall. Algoritmen ble utviklet av Professor Ronald (Ron) Rivest i 1991.

<http://no.wikipedia.org/wiki/Md5>

MySQL

MySQL er et SQL-basert databaseadministrasjonssystem som er lisensiert under GPL. Denne databasetjeneren er veldig mye brukt, og er en vesentlig del av LAMP-systemer, hvor M-en står for nettopp MySQL.

<http://no.wikipedia.org/wiki/MySQL>

Nettleser

En nettleser (engelsk web browser) er et program som vanligvis brukes til å vise innhold fra internett som oftest finnes i HTML-formatet. Nettleserens jobb er å «oversette» HTML-kodene som angir plassering av tekst og eventuelt bilder, filmer og lignende og vise dette i henhold til koden. Nettlesere opererer vanligvis på den delen av internett som kalles Verdensveven (WWW -- World Wide Web).

<http://no.wikipedia.org/wiki/Nettleser>

Open Source

Åpen kildekode (oftest omtalt med lånebegrepet Open Source fra engelsk) betyr at kildekoden til et dataprogram er gjort tilgjengelig (ofte på Internett) for alle. Det finnes mange forskjellige lisenser for åpen kildekode, men den mest brukte er GNU General Public License (GPL).

http://no.wikipedia.org/wiki/Open_source

Paging

Viser til operativsystemer som har hovedminne delt opp i sider (noen ganger kalt swapping). Paging er en overføring av sider mellom hovedminne og en lagringsplass, som for eksempel en hard disk. Prinsippet er det samme for flere sider på web.

<http://en.wikipedia.org/wiki/Paging>

Php

PHP er et dynamisk, tolket og løst typet programmeringsspråk hovedsakelig brukt for å utvikle dynamiske nettsider. PHPs syntaks ligner C og Perl. Den vanligste implementasjonen av PHP er en fri og åpen versjon skrevet i C og distribuert av The PHP Group via php.net og SourceForge. En annen fri og åpen implementasjon, som muliggjør kompilering av PHP-kode til maskinkode, er utviklet av Roadsend Inc.

<http://no.wikipedia.org/wiki/Php>

Prosessering av XML

API-ene (Application Program Interface) mest brukt for prosessering av XML-data via et programmeringsspråk er SAX og DOM. SAX brukes for sekvensiell prosessering, mens DOM brukes for uthenting av enkeltverdier.

<http://no.wikipedia.org/wiki/XML>

Return of investment (ROI)

ROI er en metode for å måle hva en har tjent eller tapt på en investering, eksempel en fortjeneste på 50 på en investering på 1000 gir følgende ROI $50/1000 = 5\%$ En fortjeneste på 20 på en investering på 100 gir ROI $20/100 = 20\%$. Dette indikerer at metoden må benyttes kritisk, ikke egnet til alle formål.

http://en.wikipedia.org/wiki/Return_on_investment

Scrolle

Med data grafikk og televisjon så er scrolling eller text crawling en måte å gli presentasjonen horisontalt eller vertikalt. Brukes til for eksempel tekst, tegninger, bilder. Brukes ofte til å vise større mengder med innhold enn det er plass til i skjermbildet.

<http://en.wikipedia.org/wiki/Scrolling>



Server

Er en programvare som tilbyr en eller flere tjenester til andre datamaskiner (klienter) over et datanettverk. Begrepet tjener/server er også ofte brukt om maskinvaren som programmet (eller programmene) kjøres fra. For eksempel kan en og samme datamaskin tilby flere tjenester samtidig, så fremt maskinen har kapasitet til å utføre alle oppgavene.
<http://no.wikipedia.org/wiki/Server>

Skript

Skriptprogrammeringsspråk, som oftest kalt skriptspråk, er en type programmeringsspråk som tolkes og utføres av separate programmer. Et skript er en fil som inneholder kommandoer i et skriptspråk. Et skript er som et «manuskript» for tolkeprogrammet. De første typiske skriptspråkene ble ofte kalt «jobbkontrollspråk».
<http://no.wikipedia.org/wiki/Skriptspråk>

SQL

Structured Query Language (SQL) er et språk som benyttes til å formulere og kjøre operasjoner mot databaser. De fleste av dagens databasesystemer tilbyr SQL som kontrollgrensesnitt. SQL definerer flere operasjoner og disse kan sorteres inn i tre grupper:

- Strukturelle operasjoner
- Operasjoner mot data
- Kontrolloperasjoner

<http://no.wikipedia.org/wiki/SQL>

Søkemotor

Søkemotor er en betegnelse for en type programvare som leter frem nettsider på internett som samsvarer med et gitt søk, og rangerer disse etter hva den oppfatter som mest relevant. Typisk ligger søkemotoren tilgjengelig som et nettsted, der brukeren legger inn søkeord, og treffet vises som klikkbare lenker.

Søkemotoren kan enten gjøre søk på hele Internett (for eksempel Google og Yahoo!), innenfor et bestemt nettsted (for eksempel søk innenfor VGs nettavis), eller innenfor et bestemt tema (f.eks. Kelkoo som søker i priser på produkter, Picsearch som søker bilder).

<http://no.wikipedia.org/wiki/Søkemotor>

XHR

Xmlhttprequest er en API som kan brukes av JavaScript og andre til å overføre data mellom en tjener og klient. XHR er en viktig del av Ajax. Data mottatt fra et XMLHttpRequest kall blir ofte tatt hånd om av en back-end database. I tillegg til XML så kan xhr brukes til å ta inn data fra andre formater som for eksempel HTML, JSON eller ren tekst.
<http://en.wikipedia.org/wiki/XMLHttpRequest>

XML

XML (Extensible Markup Language) er en W3C-anbefaling for å strukturere data eller beskrivelser av data (metadata) i elementer ved å bruke tekstkoding eller markeringskoder kalt tagger.

XML er utvidbart, hvilket betyr at det ikke finnes ett sett med elementer som alle må bruke. Man kan, hvis man ønsker det, definere sine egne elementer. Dette i motsetning til HTML og XHTML, som er «ferdige» markeringsspråk laget for et spesifikt formål.

<http://no.wikipedia.org/wiki/XML>



4. Oversikt over grunnleggende forretningsmodeller på internett

Videre redegjørelse om grunnleggende forretningsmodeller på internett er basert på en artikkel skrevet Dr. Michael Rappa og fritt oversatt til norsk og publisert av Bråthen(2003).

Abonnement

Brukeren betaler for tilgang til nettstedet eller tjenesten. Innholdet må ha svært høy verdi for brukeren. [ex: Wall St. Journal, Consumer Reports].

Generisk nyhetsinnhold, som i aviser, har vist seg å være vanskelig å ta betalt for på internett. [ex: Slate]. En studie fra Jupiter Communications i 1999 fant at 46 % av internettbrukerne ikke var interessert i å betale for nyheter på internett. Noen leverandører har kombinert gratis innhold (for å drive opp volum og trafikk) med spesialisert innhold og tjenester på abonnementsbasis.

Affiliate / Salgsmarkedsføring

I kontrast til generelle portaler som må ha høy trafikk så tilbyr affilietmodellen kjøpsmuligheter uansett hvor kunden surfer.

Salgsmarkedsføring og affiliateprogrammer gjør at e-handelssteder, portaler og informasjonsnettsteder kan oppnå økt salg og trafikk, og få kvalifiserte leveranser og nå videre ut med merkevaren, mens de bare betaler for resultatene av sine markedsføringsfremstøt. Ved å bare betale for resultater, får disse nettstedene lavere kundeanskaffelseskostnader og styrker kraften til sitt markedsføringsbudsjett.

Forretningsmodellen ligger i å tilby finansielle insentiver (% av omsetning) til partnernettsteder. Modellen er skreddersydd for internett og har derfor blitt enormt populær. (ex: TradeDoubler , BeFree) Les også i-revenue.net en guide til affiliateprogrammer på nettet, eller les AffiliateWorld

Annonsering

Modellen for nettbasert annonsering er en utvidelse av den tradisjonelle mediebaserte kringkastingsmodellen. Kringkasteren er i dette tilfellet et nettsted som tilbyr informasjon (innhold), vanligvis men ikke alltid gratis, og tjenester som e-post, chat og temabaserte fora, mikset med bannerannonser

Bannerannonser kan være den viktigste eller den eneste inntektskilden for kringkasteren. Kringkasteren kan selv skape innhold eller videreformidle innhold skapt andre steder (eller av andre). Annonsemodellen fungerer bare dersom det er høy trafikk på nettstedet eller at nettstedet er svært spesialisert.

Generell portal:

-- Høyt trafikkvolum -- typisk mange millioner per måned -- drevet av generisk eller variert innhold eller tjenester. (ex: søkemotorer eller kategoriserte oversikter som Excite, AltaVista og Yahoo! eller innholdsdrivne steder som AOL). Det høye volumet gjør annonsering attraktivt og forhåpentligvis lønnsomt. Dette gir mulighet for mer innhold og flere tjenester. Konkurransen om trafikk har ført til omfattende tilbud av gratis innhold og tjenester som e-post, nyheter, chatområder, lokal informasjon etc.



Personalisert Portal:

-- Den generiske modellen til generelle portaler underminerer brukerlojalitet. Dette har ført til nye typer portaler (ex: My.Yahoo!, My.Netscape) som tillater brukertilpassing av grensesnitt og innhold. Dette øker kundelojaliteten dramatisk siden brukeren benytter mye tid på (i) portalen, blant annet for å tilpasse den til egne behov. Lønnsomheten er basert på volum og til en viss grad den informasjonen brukeren velger å lese. En personalisert portal kan støtte en "spesialisert portal".

Spesialisert Portal – Vertikal:

-- Her er volum mindre viktig enn en veldefinert brukergruppe (kanskje 0.5-5 millioner besøk per måned). Dette er nettsteder som eksempelvis retter seg mot golfere, boligkjøpere, nybakte foreldre og andre interessegrupper. Disse er attraktive for målrettet reklame og affiliateprogrammer. Det er grunn til å tro at spesialiserte portaler vil få en sterk vekst. Det er også grunn til å tro at spesialiserte portaler blir tjenester i elektroniske markedsplasser.

Oppmerksomhets- / Insentiv markedsføring:

-- En markedsplass som bygger på ideen om at selv om tilgjengelig informasjonsmengde er uendelig så er tilgangen begrenset av knapphet på tid hos brukeren. Oppmerksomhetsøkonomi har eksistert så lenge medier har vært kommersielle. Kommersielle medier sitte egentlige produkt er ikke å være magasiner men å skape blikkfang for annonsører. Interaktive medier tar dette konseptet ett skritt lengre og bygger på at trafikk kan kjøpes, selges eller byttes og øyeblikkelig sendes videre til andre nettsteder.

Alt kan skaleres opp til en milliard mennesker som ser på olympiske leker, eller ned til din lille målgruppe. Oppmerksomhetsøkonomien forklarer den eksplosive veksten av svært synlige nettsteder som Yahoo og profileringen av gratisprodukter (for brukeren) og -tjenester. I en internettverden med stadig mer og mer trafikk blir denne type verktøy for å skaffe og holde på oppmerksomhet mer og mer verdifull.

Gratismodellen:

-- Gir brukerne noe gratis : drift av hjemmeside [ex: FreeMerchant], internettaksess, elektroniske julekort [BlueMountain]. Gratisprodukter genererer mye trafikk som igjen skaper muligheter for annonsører. Lønnsomhet er et problem når den kun er basert på annonsering. Modellen kan med fordel kombineres med infomediary modellen.

Rabattmodellen:

-- Det beste eksempelet er Buy.com, som selger produkter til under selvkost og hvor lønnsomhet er basert på annonsesalg.

Brokerage / Megler

- Meglere og markedstilretteleggere

De kobler kjøpere og selgere og fasiliterer transaksjoner. Dette kan være innenfor B2B (business-to-business), B2C (business-to-consumer), eller C2C (consumer-to-consumer). En megler tjener penger på en avgift per transaksjon. Meglerbaserte modeller kan være av ulike typer:



Kjøps-/Salgsgjennomføring:

-- Dette kan være en finansiell megler som eTrade, hvor kunden legger inn kjøps- og salgsordrer for finansielle instrumenter som eksempelvis aksjer. Reisebyråer faller også inn under denne kategorien og også her påløper det en transaksjonsavgift for enten selger eller kjøper. Noen modeller er basert på volum og lav overhead for å levere de laveste prisene. Ett eksempel på dette er CarsDirect .

Markeds plass:

-- Svært vanlig i B2B markedet. Gode eksempler er MetalSite eller ChemConnect's World Chemical Exchange . I markeds plassmodellen belastes vanligvis selgeren en transaksjonsavgift basert på den pris som er oppnådd i salget. Prismekanismen kan være tilbud/kjøp, tilbud/forhandling eller auksjonsbasert.

Kjøpersentrert:

-- Modellen ble lansert av Accompany , som beskriver kjøperaggregering som prosessen med å samle kjøpere på internett slik at de kan opptre som en gruppe, og gjennom dette oppnå de samme innkjøpsprisene som store organisasjoner som kjøper inn i store volum. Selgerne betaler en liten prosentandel av salget på transaksjonsbasis.

Distributør:

-- En katalogbasert modell hvor mange leverandører og produsenter blir koblet med grossister og detaljister. B2B modeller er svært vanlig. Meglere fasiliterer transaksjoner mellom distributører og deres handelspartnere. Kjøperne kommer raskere ut i markedet og får raskere tilgang på nye produkter samtidig som innkjøpskostnadene blir lavere. Gjennom å tilby kjøperne muligheter for rabatter fra foretrukne leverandører, tilgang til kjøperspesifikke priser, ledetider og alternative produkter blir transaksjonen mer effektiv. For distributørene reduseres også salgskostnadene ved at både salg, ordrebehandling og endringshåndtering blir mer effektiv. [ex: NECX]

Virtuelt varehus

-- Et nettsted som inneholder mange detaljister/nettbutikker. Disse blir belastet for oppsetting av nettbutikk, annonser og/eller gjennom transaksjonskostnader. [se eksempelvis Yahoo! Store's terms. Modellen er mest effektiv når den kombineres med en generell portal. Mer sofistikerte varehus vil også tilby betalingstjenester og CRM muligheter for sine "leietagere" [ex: Yahoo! Stores, ChoiceMall, Women.com's Shopping Network]

Metamediary

- En metamediary fungerer som et sentralbordet mellom kunder. Grunnkonseptet for B2B markeds plasser er at de kobler kjøpere med selgere over hele kloden. Disse kunne ikke funnet frem til hverandre uten denne type markeds plasser. Her kan de nå utveksle informasjon, utnytte markeds plassens og hverandres tjenester og selvsagt gjennomføre transaksjoner.

Det er en variant av et virtuelt varehus, men den tar i tillegg ansvar for gjennomføring av transaksjonen, sporer/følger ordrer og tilbyr fakturerings- og betalingstjenester. En metamediary opptrer på vegne av de som selger og står som garantist i relasjonen mot alle som kjøper. Også her belastes selgerne (butikkene) med en oppsettingsavgift og transaksjonsavgifter. Virtuelle varehus vil sannsynligvis bevege seg i denne retningen.



Det er i det utvidede tjenestetilbudet metamediaries har sin rolle. Kredittkorthåndtering, produktprising, leveransedata, kundetilfredshetsdata, kvalitetssikring er noen av tjenestene som både selgere og kjøpere skal verdsette.

"Metamediaries representerer ikke leverandørene men fasiliteter markedsplassen og dens tjenester. De deltar ikke selv i transaksjonene, men tjener penger på å koble kjøpere og selgere" sier Professor Sawhney. Metamediary er et varemerke fra Kellogg Graduate School of Management assistant professor Mohan Sawhney.

Auksjonstilrettelegger (Megler / Broker)

-- Et nettsted som gjennomfører auksjoner for selgere (individuelle eller forretninger). Selgeren blir belastet en avgift som beregnes på salgsværdien. Selger aksepterer høyeste bud over minstepris. Auksjoner kan ha ulike regler for tilbud og budgivning. [ex: eBay, AuctionNet, Onsale]

Motsatt auksjon

-- Modellen ble etablert og patentert av Priceline. Potensielle kjøpere gir et bindende bud og megleren finner selgere som er villig til å selge til denne prisen. Avgiften beregnes på forskjellen mellom bud og endelig pris samt et eventuelt administrasjonsgebyr for gjennomføring av transaksjonen.

Betalt annonsering - Rubrikkannonser

-- Oversikter over ting for salg eller ting som ønske kjøpt. Drives normalt av innholdsleverandører. Annonseren faktureres uavhengig om annonsen fører til salg eller kjøp. [Ex: Finn]

Søkemotor (Agent)

-- En agent (eller robot) som benyttes for å finne den billigste prisen på et produkt eller tjeneste du ønsker å kjøpe. Kan også benyttes for å finne fremt til informasjon som ellers er vanskelig å finne. [Ex: Gulepriser, DealTime, MySimon, RoboShopper, R U Sure, ShopFind] Et nettsted for jobbformidling kan fungere som en søkemotor for bedrifter som leter etter medarbeidere eller jobbsøkere som leter etter arbeidsgiver. [Stepstone, aEtat]

Community / Nettverk / Elektronisk samfunn

Suksesskriteriet for denne modellen er kundelojalitet (i motsetning til høy trafikk). Brukerne bruker mye tid per besøk, har et emosjonelt forhold til nettstedet og ofte er de også aktive innholdsprodusenter. Modellen kan også være abonnementsbasert for spesielle tjenester.

Handelsnettverk:

-- eller "vertikale samfunn" et konsept unnfanget og igangsatt av VerticalNet. Dette er nettsteder som posisjonerer seg som "betydningsfullt og med omfattende og god informasjon om et spesifikt markedssegment eller industrisektor" VerticalNet's handelsnettverk inneholder produktinformasjon i kjøpsguider, leverandør og produkt databaser, daglige industrinyheter og artikler, jobbanonser og målrettet reklame. I tillegg har VerticalNet integrert en rekke B2B markeds plasser som supplerer tradisjonelle markedsaktiviteter. [se også: Buzzsaw.com]

Kunnskapsnettverk:



-- eller ekspertnettverk som tilbyr informasjon og utveksling av profesjonell ekspertise og erfaring. Disse nettstedene er ofte tilrettelagt som fora hvor deltagerne kan stille spørsmål og få svar fra andre deltagere. Ekspertene kan være ansatt på nettstedet eller ganske enkelt andre brukere som ønsker å svare. [ExpertCentral, AskMe, Abuzz] Også avgiftsbaserte modeller [Ex: Guru, Exp]

Infomediary / Informasjonsmellommenn

Data om kunder og deres kjøpsmønstre er informasjon med høy strategisk verdi. Spesielt når denne informasjonen har vært gjennom en grundig analyse og kan benyttes for målrettet markedsføring.

Noen selskaper har dette som forretningsmodell og lever av å samle, analysere og selge informasjon om kunder. En informasjonsmellommenn kan tilby brukere gratis internetttilgang [NetZero] eller gratis PC/mobiltelefon [eMachines.com] i bytte med detaljert informasjon om deres kjøpsmønstre og internettbruk. Dette er en modell som ser ut til å ha større gjennomslagskraft enn den rene annonsemodellen (bannerannonser etc.)

Modellen kan også benyttes motsatt vei og gi kundene nyttig informasjon om nettsteder i et markedssegment som sloss om kundenes penger - hvilket nettsted skal jeg benytte for å selge mine produkter / tjenester.

Kriterier:

En infomediary, (informasjonsmellommenn), tilbyr informasjon og tjenester du ikke får andre steder eller på annen måte og hvor :

Internet er både leveransekanal og markedsføringsmedium.

Tjenesten kan avslutte transaksjonen, sørge for gjennomføring av transaksjonen, spore transaksjonen i tillegg til å starte den.

Markedet har kritisk masse av kunder eller potensiell kunder.

Person-til-person interaksjon er eliminert.

Det er lett å gå inn i nye markeder gjennom å tilby parallelle tjenester.

Informasjonen som samles inn er lett å reselge, repakke og gjenbruke.

Anbefalingsmodellen

-- Dette er et nettsted hvor brukerne kan utveksle informasjon og dele erfaringer om produktkvalitet eller tjenestekvalitet -- eller om den de har kjøpt produktene eller tjenestene fra. [Se: ePinions].

Modellen kan kombineres med affiliasmodellen og hever da verdien på dataene betydelig.

Registreringsmodellen

-- Innholdsbaserte nettsteder som er gratis å benytte, men som krever registrering for bruk. Annen informasjon kan også tenkes å bli samlet inn (Ved å besøke et nettsted legger du alltid igjen noen spor. Eks hvor du kommer fra). Registrering åpner for å bygge opp kunnskap om brukerens bruksmønster på nettstedet. Slik informasjon har stor verdi i målrettede annonsekampanjer. Dette er basismodellen for en infomediary eller informasjonsmellommenn. [ex: NYTimes.com]

Utility / Betal etter forbruk



Modellen måler forbruk og avregner / tar betalt etter hvor mye tjenesten benyttes. Suksessen er avhengig av muligheten til å ta betalt "per Byte" og bør inkludere mulighet for mikrobetaling (beløp som er for små til å betales med kredittkort eller faktura) [ex: Authentica].



5. Informasjon til informant i forkant av brukertest

Før du starter testen.

Som et ledd i vår utdanning på handelshøgskolen i Bodø er vi tre studenter som gjennomfører et utviklingsprosjekt for en ny internettløsning. For å få til en best mulig løsning for de som skal bruke denne nettsiden trenger vi synspunkter fra personer som ikke kjenner løsningen i utgangspunktet.

Intensjonen med å testen denne løsningen er å få avdekket svake sider i en tidlig utviklingsfase. Derfor vil alle tanker, synspunkter og spørsmål være av stor interesse og nytte for oss. Vi ber derfor om at du spør og kommenterer underveis når du prøver å navigere på nettsiden.

Dine bevegelser på skjermen vil bli registrert i en videofil i forbindelse med testen. Men så snart vi har fått dokumentert testen i rapportform vil denne videofilen bli slettet.

Presentasjon av testresultater vil gjøres uten å avsløre noe om testpersonene.

På forhånd takk

Forbrukertipset.no



6. Dokumentstruktur på XML data fra Tradedoubler.

Eksempel på produkt feed I xml format fra TradeDoubler. Filen viser data for kun et produkt.

```
<?xml version="1.0" encoding="UTF-8"?>
<products>
  <product>
    <name>Dell™ Laser Printer 1110</name>
    <productUrl>http://pdt.tradedoubler.com/...</productUrl>
    <imageUrl>http://i.dell.com.....</imageUrl>
    <description>Dell&#153; laserskriver 1110</description>
    <price>895.38</price>
    <currency>NOK</currency>
    <TDProductId>65289153</TDProductId>
    <TDCategoryID>25</TDCategoryID>
    <TDCategoryName>Printere</TDCategoryName>
    <merchantCategoryName>Laserprinter</merchantCategoryName>
    <sku>™ Laser</sku>
    <shortDescription>Dell 1110-laserskriver gir...</shortDescription>
    <promoText/>
    <previousPrice>1198.75</previousPrice>
    <warranty/>
    <availability>På lager</availability>
    <inStock/>
    <shippingCost>Not Available</shippingCost>
    <deliveryTime/>
    <weight/>
    <size/>
    <brand/>
    <model/>
    <ean/>
    <upc/>
    <isbn/>
    <condition>New</condition>
    <mpn/>
    <techSpecs/>
    <manufacturer/>
    <programName>Dell NO</programName>
    <programLogoPath>http://img.tradedoubler...</programLogoPath>
    <programId>1410</programId>
    <fields>
      <ArticleID>1110</ArticleID>
      <Channel>dhs</Channel>
      <HomeDelivery>Yes</HomeDelivery>
      <Processor>Dell™ laserskriver 1110</Processor>
      <productLink>http://configure2.euro.dell.com...</productLink>
    </fields>
  </product>
</products>
```

7. PHP kode for serversiden av prototyp. Skript: fillMainSubCatDb.php

```

<?php
/*****
| Skriptet fyller hoved- og sub kategori tabellene med data fra array
| *****/

$mmainSubCat=array(
'Ukategorisert'=>array('Ukategorisert'),
'Data og IT'=>array('Annen lagring','Bredbånd og ISP','Bæresesker data',
'CD/DVD/Bluray stasjoner','CPU','Diverse data og it','Minne','Modem',
'Nettverk','PC Stasjoner','PC Bærbar','PDA','Plate lagring','Programmer',
'Sikkerhet','Skannere','Skjerm','Skjermtilbehold','Skrivere',
'Stromforsyning','Tastatur/Mus','Tilbehold bærbar','Tjenere',
'Tuner/video kort','USB Periferiutstyr'),
'Erotikk'=>array('Diverse erotikk','Erotisk film','Erotiske leketøy',
'Erotiske spill og bøker','Erotiske undertøy/klær'),
'Film, musikk og litteratur'=>array('Blader og tidsskrift','Bøker','Film',
'Musikk'),
'Foto, video og audio'=>array('A&V kabler','Kamera','Kameratilbehold',
'Mp3 spiller','Mp3 tilbehold','Musikkinstrument',
'Presentasjon og konferanse','A&V Skjermtilbehold'),
'Helse, velvære og trening'=>array('Helse','Kroppspoleie','Sminke','Trening'),
'Hjem og bolig'=>array('Diverse hjem og bolig','Husdyr','Kikkert',
'Kjekkenapparater','Kjeleskap','Kontaktlinser','Mikrobelgeovn',
'Oppvarming','Støvsugere','Vaskemaskiner'),
'Lyd og bilde'=>array('DVD','Fjernkontroller','Hjemmekino','Høytalere/headsett',
'Projektor','Radio','Tilbehold projektor','Tuners','TV','Video'),
'Mobil'=>array('Abonnement','Diverse mobil','Mobil med abonnement',
'Mobil uten abonnement','Ringetoner','Tilbehold mobil',
'Mobil underholdning'),
'Spill og leker'=>array('Diverse spill og leker','Brettspill','Dataspill',
'Leker','Spillmaskiner')
);

$mmainTable='pr_mainc';$subTable='pr_subca';
$mmainFields=array('maincName','maincDescript');
$subFields=array('id_mainc','subcaName','subcaDescript');
$mmainId=0;
openDb();
foreach ($mmainSubCat as $mmain => $sub) {
    $mmainId++;
    $mmain=cleanUp($mmain);
    queryDb("INSERT INTO $mmainTable
    SET $mmainFields[0]='$mmain',$mmainFields[1]='$mmain'");
    foreach ($sub as $key => $subVal) {
        $subVal=cleanUp($subVal);
        queryDb("INSERT INTO $subTable
        SET $subFields[0]='$mmainId',$subFields[1]='$subVal',
        $subFields[2]='$subVal'");
    }
}
closeDb();

function cleanUp($str){
    $str=trim($str); #Fjerner "whitespace"
    $str=htmlspecialchars($str, ENT_QUOTES, 'UTF-8'); #Endrer spesialtegn
    return $str; #Returnerer rensket streng.
}
?>

```



8. PHP kode for serversiden av prototyp. Skript: dataToLinkTab.php

```

<?php
/*****
| Skriptet fyller databasetabell med data fra en xml fil fra Tradedoubler
| Tabellen skal fungere som en link tabell mellom sub-kategori og produkt
| *****/
$dbTab='cat_prod_td';          #Databasetab. som skal skrives til

openDb();
  parseToDb($dbTab);          #leser xml fil og henter ut data.
closeDb();

function parseToDb($dbTab){
  $tag=array( 'name','TDCategoryName',
              'merchantCategoryName');
  $xmlFile='../data/tips/pf_284570.xml';
  $feed = XMLReader::open($xmlFile);
  $nam=''; $stdC=''; $mer='';
  while ($feed->read()) {
    switch ($feed->nodeType) {
      case XMLREADER::ELEMENT:
        if ($feed->name==$tag[0]){
          $feed->read();
          $nam=$feed->value;
          $nam=cleanUp($nam);
        }elseif ($feed->name==$tag[1]){
          $feed->read();
          $stdC=$feed->value;
          $stdC=cleanUp($stdC);
        }elseif ($feed->name==$tag[2]){
          $feed->read();
          $mer=$feed->value;
          $mer=cleanUp($mer);
        }else{
          break;
        }
      case XMLREADER::END_ELEMENT:
        if ($feed->name=='product') {
          dbInput($dbTab,$stdC,$mer,$nam);
        }else{
          break;
        }
    }
  }
  $feed->close();
}

function cleanUp($str){
  $str=trim($str);
  $str=htmlspecialchars($str, ENT_QUOTES, 'UTF-8');
  return $str;
}

function dbInput($dbTab,$stdC,$mer,$nam){#Skriver en rad til databasetabell.
  $id=md5($stdC.$mer);
  if (idExists($dbTab,$id)==false) {
    queryDb("INSERT INTO $dbTab SET id='$id', subcat='', trdcat='$stdC',
          mercat='$mer', prname='$nam'");
  }
}

function idExists($dbTab,$id) {
  $result=queryDb("select count(*) from $dbTab where id='$id'");
  return fetchValueDb($result);
}
?>

```



9. PHP kode for serversiden av prototyp. Skript: addCatToLinkTab.php

```

<?php
/*****
| Scriptet fyller inn data i kategorifeltet i databasetabell fra array
| Tabellen skal fungere som en link tabell mellom sub-kategori og produkt
| *****/
$dbTab1='cat_prod_td';           #Databasetab. som skal skrives til.
$dbTab2='pr_subca';            #Databasetab. med subcategory ID.

openDb();                       #Åpner databasen.
foreach ($md5CatProd as $md5 => $cat) { #Looper gjennom kategori array.
    //Henter kategori id tilsvarende array verdi(kategori).
    $result=queryDb("select ID from $dbTab2 where subcaName='$cat'");
    $row=fetchRowDb($result);
    $subcaId=$row['ID'];
    //Oppdaterer kategori id for rad som tilsvarende array nøkkel(md5).
    queryDb("UPDATE $dbTab1 SET subcat='$subcaId' WHERE id='$md5'");
}
closeDb();                       #Stenger databasen.

$md5CatProd=array(              #Md5 hash tilordnet våre kategorier.
    '06e8b2e1dc7fbb667468dc20870f8cb4'=>'Ukategorisert',
    '12cb1056c51d8929fd03b2f209bd8bf3'=>'Ukategorisert',
    '6d73917712608d4b6035f63274c03009'=>'Mobil underholdning',
    '3a5888d7444cedc135d3dd0df7535f2e'=>'PC Bærbar',
    'ebb0c9f5efa4ebb6fd94ecd2a366ad8a'=>'PC Bærbar',
    'a0071641a6f89d173d6aeff5b95359a0'=>'PC Bærbar',
    '722496160d9e02ce30518f38b99a43ba'=>'Mp3 tilbehør',
    '9e756a049c7c320d38dc1c731c287d38'=>'Mp3 tilbehør',
    '0de9b241123ee60775022959c8d3001e'=>'Mp3 tilbehør',
    'a7ab23ef2fc6f81ef4c892eae71398eb'=>'Mp3 spiller',
    '7dbc8058afe839c44cb3e04f4d2dce4b'=>'Mp3 spiller',
    'aefd8217d10070d8147d1800959e9905'=>'Mp3 tilbehør',
    '927d2f312ce3b7195899da636176b922'=>'Bøker',
    '3807adb6e73d5092653f7bb62f3c939e'=>'Blader og tidsskrift',
    'ee9df12fb3332c95077d7e96e1f6calf'=>'Bredbånd og ISP',
    'a3a4325100e8008449fe211c707d715f'=>'Musikk',
    'f8303e100b8b67aafebe99501552e749'=>'Ukategorisert',
    '95819f346b62abdabb183a966fa8de48'=>'Ukategorisert',
    'b28677c89c45b1edc013c734defdlfb0'=>'Ukategorisert',
    'a3da12a115b5861a70e7170a7a96d2f0'=>'CD/DVD/Bluray stasjoner',
    '3ac3770a99dfe4939b5fda5667712f3d'=>'CD/DVD/Bluray stasjoner',
    '8550de76d33566c0c3d18dbb96c35bd4'=>'Mobil underholdning',
    '2788188fc8ee7a59f1016d4f90cacc14'=>'CPU',
    '284162121b2feb83966c59860724005b'=>'Bøker',
    '4019cbe548010c30b8085e1757ed95c7'=>'Dataspill',
    '7330ed22e40c978f26c714236f3a2e5f'=>'Skrivere',
    'bc4ef173ba17e7d8068d33d03336e47c'=>'Skrivere',
    'ce121d686daa2c0f669855ba95339a83'=>'Skrivere',
    '5058e9c65620f8a633355dbd2375ad11'=>'Radio',
    '44b5a3e5f8c39e8dd3eeef04c3a9990e'=>'CD/DVD/Bluray stasjoner',
    'd815185a03823ee4cfcdea8cfa7d13a4'=>'Film',
    'd79b232cb824ed82a7d376c2dd12b8f1'=>'Film',
    '39f6b8ff78918b3e43fe0ac4e2787f42'=>'Film',
    '1e2656eeefb3c7c2cf2d12a21be2a407a'=>'DVD',
    'f31f355e2255f047259d2e72182a75b5'=>'Tastatur/Mus',
    '87ef6df9457fa32f9fc25f31eb6a8972'=>'Kjøkkenapparater',
    '72f4fafcde551f6dac34be2af508b827'=>'Kjøkkenapparater',
    '95565a2668b6ac1a10be4a37705eff49'=>'Diverse hjem og bolig',
    'ac6740c80a5ce3f183e0c673170688af'=>'Diverse hjem og bolig',
    '10cbba66b63d838edc60f6bbccdeac65'=>'Diverse hjem og bolig',
    'b8d5604f313d26c1f784e0dalb3e0540'=>'Erotisk film',
    '9b05f5382fe647519bb1944995ffd8c1'=>'Erotisk film',

```



```
'8c0dcb48337e6db3f3a6040af532e001'=>'Programmer',
'c00ccf4cf4f9df450c84467a493e474f'=>'Prosjektor',
'c9ca830168ba22a75547celf1e14f1e7'=>'Prosjektor',
'73bc7fafc655868dcb8782ba392214a'=>'Ringetoner',
'6073c036b51535b3b8368abfcbde27eb'=>'Sikkerhet',
'bb73cd717a1d0be41df26ba6e8347764'=>'Sikkerhet',
'0d226843da94c4e86073acc92f46e10e'=>'Sikkerhet',
'ba9c90256c074017f797a36a9f4d2da6'=>'Sikkerhet',
'03f360ac8de9cca5e7b908a706e0fdfa'=>'Sikkerhet',
'b67fd7edf74dca822aa72611927c3063'=>'Sikkerhet',
'33bbfe9c57c3a7b3b0510dfc69dc8f5c'=>'Sikkerhet',
'b0c3f8b89bbb37be615a143eebf42e6e'=>'Sikkerhet',
'cbcc88188d7b195b6afe89b7abe325ed'=>'Sikkerhet',
'6713dfe9ecb0bbb162684212afcc662f2'=>'Skannere',
'955fae55720a155de4a08359728e7943'=>'Skannere',
'fafcc85e81aa5204382f7676b5c07f87'=>'Skannere',
'20ccc8ddc2d1307cbf7c885d45aa47f4'=>'Skjerm',
'54abb551b24a40aa9975a9ded2303d50'=>'Skjerm',
'c9888979f2b1b319cc27fde3aaece01f'=>'Skjerm',
'4ccca90496719fad3961660b6adf5e14'=>'Skjerm',
'blee49c2e1152ccd7e70ef5a62096ca4'=>'Skjerm',
'3d2690e88efd21bc38827234e865c1df'=>'Skjerm',
'c861163586938bca035d0edb710dedc7'=>'Skjermtilbehør',
'cela72314206f43c9ba6c74aeb337bbe'=>'Skjermtilbehør',
'684f9d56ba74846993f309beb9e30531'=>'Skjermtilbehør',
'e4428f2f2234cdc3e20d3fbacd2fe426'=>'Skjermtilbehør',
'6e70cd5787b8cdd4a9dcca6138d2f232'=>'Hjemmeunderholdning',
'6b8536ae153899f1b12053c3487e8256'=>'Video',
'4ca6a551269b306b58cde56cafec966'=>'Sminke',
'0512bb52904d687440febf1cc37a9c36'=>'Sminke',
'9822c066d1f578d5a6c124737ae584f8'=>'Trening',
'53b3278fe301eabf61ef0a597a5c4a73'=>'Ukategorisert',
'4f7e7dccc1f7cb28ac00a239ffcdffe67'=>'Støvsugere',
'9cbf76da3bb918b71dela2a822c6b38d'=>'PC Stasjonær',
'1779c9723f48c0103b04483f89516dc2'=>'PC Stasjonær',
'3508331c15a5724e8be7fc2e81091194'=>'PC Stasjonær',
'63c012e351e81c78f01c56a68bbf9e1b'=>'PC Stasjonær',
'05a1bef421b409c6f2f39425ef24cb51'=>'PC Stasjonær',
'd490f4e13de061434af9c9d73ab2415f'=>'Strømforsyning',
'4de03bba56fed8144a133e4d3589ab7d'=>'Strømforsyning',
'32815f2biad456b6c078d67ca7b41703'=>'Mobil underholdning',
'072816e692ee32e3f5ff10e8af98ffe5'=>'Tilbehør',
'c295f5987da9b55ffdblab81125dedc0'=>'Diverse hjem og bolig',
'ffcef664df5b4b99d4a3f8094ad70291'=>'Tjenere',
'f00b180f60ef44783a10ee9d6a9fb8f2'=>'Tjenere',
'a2e32de697bdd688576ce8dcd8f682f0'=>'Trening',
'4c7e40b7743cca0ec6f80b2e23f78022'=>'Tuners',
'504f473832000d5d52473ad336624b79'=>'USB Periferiutstyr',
'43f76b269c3c6f3af0cf5510818db2f5'=>'Ukategorisert',
'86ef85239b531c8e8fa3f8fa8fca7c0e'=>'Vaskemaskiner',
'9a71633e3edec7e0e651cf379326a2f7'=>'Vaskemaskiner',
'e2b0806c85037c8aa3ef1d53157dd3a5'=>'Vaskemaskiner',
'dfd0076738bff3187e8b23541cb1813c'=>'Barevesker data',
'059a4454b9e97b4a35301e69ed9365ba'=>'Barevesker data',
'73560125e0d4f597ee32ab371427dad1'=>'Barevesker data',
'128103ea3b845120d54e37a4a15c48ba'=>'Kabinett',
'6338053e37753d3606321c1aef6b23e'=>'Mobil underholdning',
'479449b51396cc772068f7e5958912b1'=>'Spillmaskiner',
'3023e8b33de492703b6a42b938cc633c'=>'Spillmaskiner',
);
?>
```



10. PHP kode for serversiden av prototyp. Skript: readToProdTab.php

```

<?php
/*****
| Scriptet fyller databasetabell med data fra en xml fil fra Tradedoubler
| Tabellen skal fungere som en link tabell mellom sub-kataegori og produkt
| *****/

/*Funksjonen leser angitte xml fil og tilordner oenskede data til variabler
for hvert 'prodct'. Disse skrives saa til databasen 'pr_produ' */
function parseToDb($xmlSource){
$tag=array(
'name', 'productUrl', 'imageUrl', 'description', 'price', 'currency',
'TDCategoryName', 'merchantCategoryName', 'brand', 'model', 'programName',
'programLogoPath'); #XML ELEMENT'ene som skal haandteres

$name=''; $pUrl=''; $iUrl=''; $desc='';
$pric=''; $curr=''; $tdCa=''; $meCa='';
$bran=''; $modN=''; $resa=''; $reLo=''; #Initierer nødvendige variabler

$feed = XMLReader::open($xmlSource); #Aapner xml-fil og tilordner $feed
while ($feed->read()) { #Mens $feed leses node for node
switch ($feed->nodeType) { #Når ->read finne nodetype[0-17]
case XMLREADER::ELEMENT: #Er node ELEMENT start-tag[1] saa
if ($feed->name==$tag[0]){ #Hvis ELEMENT tagnavn = 'name' saa
$feed->read(); #Flytt til denne tag'ens innhold
$name=$feed->value; #Tilordne innhold til egen variabel
$name=cleanUp($name);
}elseif ($feed->name==$tag[1]){ #Ellers hvis ELEMENT navn='productUrl'
$feed->read(); #Flytt til denne tag'ens innhold
$pUrl=$feed->value; #Tilordne innhold til egen variabel
$pUrl=cleanUp($pUrl);
}elseif ($feed->name==$tag[2]){ #Ellers hvis ELEMENT navn='imageUrl'
$feed->read(); #Flytt til denne tag'ens innhold
$iUrl=$feed->value; #Tilordne innhold til egen variabel
$iUrl=cleanUp($iUrl);
}elseif ($feed->name==$tag[3]){ #Ellers hvis ELEMENT navn='description'
$feed->read(); #Flytt til denne tag'ens innhold
$desc=$feed->value; #Tilordne innhold til egen variabel
$desc=cleanUp($desc);
}elseif ($feed->name==$tag[4]){ #Ellers hvis ELEMENT navn='price'
$feed->read(); #Flytt til denne tag'ens innhold
$pric=$feed->value; #Tilordne innhold til egen variabel
$pric=cleanUp($pric);
}elseif ($feed->name==$tag[5]){ #Ellers hvis ELEMENT navn='currency'
$feed->read(); #Flytt til denne tag'ens innhold
$curr=$feed->value; #Tilordne innhold til egen variabel
$curr=cleanUp($curr);
}elseif ($feed->name==$tag[6]){ #Ellers hvis ELEMENT navn='TDCat...Name'
$feed->read(); #Flytt til denne tag'ens innhold
$tdCa=$feed->value; #Tilordne innhold til egen variabel
$tdCa=cleanUp($tdCa, 1);
}elseif ($feed->name==$tag[7]){ #Ellers hvis ELEMENT navn='merch...Name'
$feed->read(); #Flytt til denne tag'ens innhold
$meCa=$feed->value; #Tilordne innhold til egen variabel
$meCa=cleanUp($meCa, 1);
}elseif ($feed->name==$tag[8]){ #Ellers hvis ELEMENT navn='brand'
$feed->read(); #Flytt til denne tag'ens innhold
$bran=$feed->value; #Tilordne innhold til egen variabel
$bran=cleanUp($bran);
}elseif ($feed->name==$tag[9]){ #Ellers hvis ELEMENT navn='model'
$feed->read(); #Flytt til denne tag'ens innhold
$modN=$feed->value; #Tilordne innhold til egen variabel
$modN=cleanUp($modN);
}
}
}

```



```

}elseif ($feed->name==$tag[10]){ #Ellers hvis ELEMENT navn='programName'
    $feed->read(); #Flytt til denne tag'ens innhold
    $resa=$feed->value; #Tilordne innhold til egen variabel
    $resa=cleanUp($resa);
}elseif ($feed->name==$tag[11]){ #Ellers hvis ELEMENT navn='pr..LogoPath'
    $feed->read(); #Flytt til denne tag'ens innhold
    $reLo=$feed->value; #Tilordne innhold til egen variabel
    $reLo=cleanUp($reLo);
}else{ #Ellers naar ingen av de over slaar til
    break; #Bryt casesloeyfen for ELEMENT
}
}
case XMLREADER::END_ELEMENT: #Er nodet ELEMENT slutt-tag[15]
    if ($feed->name=='product') { #Hvis ELEMENT tagnavn = 'product'
        newProd('pr_produ_ud',1,1,idSubca($stdCa,$meCa),
            idPricCa($pric),$name,$pUrl,$desc,
            $pric,$curr,$iUrl,$modN,$resa); #Legger til produkt i tab 'pr_produ'
    }else{
        break; #Bryt casesloeyfen for END_ELEMENT
    }
}
} #Avslutter switch betingelsen
} #Avslutter while loopen
$feed->close(); #Stopper XMLReader ob. og frigir minne
}
/*Funksjonen fjerner "whitespace" foran og bak verdi og endrer spesialtegn
til ISO entiteter og returnerer rensket tekst-streng. */
function cleanUp($str, $noMysql=0){
    if($noMysql==false){ #Ungaar behandlig av 'TDCategoryName',
        'merchantCategoryName'
        $str=mysql_real_escape_string($str);
    }
    $str=trim($str); #Fjerner "whitespace"
    $str=htmlspecialchars($str, ENT_QUOTES, 'UTF-8'); #Endrer spesialtegn
    return $str;
}
/*Funksjonen sjekker opp egendef. sub-kategori som produktet skal tilordnes
basert pa md5-hash streng av 'TDCategoryName' og 'merchantCategoryName' */
function idSubca($stdCa,$meCa){
    $linkId=md5($stdCa.$meCa); #Hash = id i tabell 'cat_prod_td'.
    if (getDbData('exist','cat_prod_td','id',$linkId)==false) {#Eksisterer id?
        $subCatId=1; #Hvis ikke blir kategori satt til 1.
    }else{ #Ellers
        $subCatId=getDbData('proCa','cat_prod_td',$linkId);#Hent hovedkat. i tab.
    }
    return $subCatId;
}

function idPricCa($pric){
    $pric=intval($pric);
    if($pric>0 && $pric<99000){
        $pricCaId=getDbData('priCa','pr_price_cat',$pric);#Hent pris-data fra tab.
    }elseif($pric=0){
        $pricCaId=1;
    }else{
        $pricCaId=30000;
    }
    return $pricCaId;
}
?>

```



11. PHP kode for serversiden av prototyp. Skript: funk_database.php

```

<?php
/*****
| Scriptet styrer den applikasjonsuavhengige MYSQL databaseh ndteringen
| *****/
require_once "script_DB_lang.php"; #Noedvendige data for database-paalogging

$queryLog=array();
$dbLink=null;

function openDb() { # pner databasen
    global $dbLink;
    $dbLink=mysql_connect(STR_DESERV, STR_DBUSER, STR_DEPASS);
    if (!$dbLink)
        die("database connect failed: ".mysql_error());
    $status=mysql_select_db(STR_DBNAME);
    if (!$status)
        die("database select failed: ".mysql_error());
}

function closeDb() { #Stenger databasen
    global $dbLink;
    mysql_close($dbLink);
}

function queryDb($query) {
    global $queryLog;
    global $dbLink;
    $queryLog[]=$query;
    $result=mysql_query($query, $dbLink);
    if (!$result)
        die("query '$query' failed: ".mysql_error());
    return $result;
}

function fetchRowDb($result) {
    return mysql_fetch_assoc($result);
}

function fetchValueDb($result) {
    if (!is_null($row=mysql_fetch_array($result)))
        return $row[0];
}

function checkDbTable($dbTab, $status=0){
    $result = queryDb("SHOW TABLES FROM ".STR_DBNAME);
    while($row = mysql_fetch_array($result, MYSQL_NUM)) {
        if ($row[0]==$dbTab) {
            $status=true;
        }
    }
    return $status;
}
?>

```




12. PHP kode for serversiden av prototyp. Skript: xmlOutput.php

```

<?php
/*****
| Skriptet som kalles fra klient og som holder styr på hvilken kolonne som
| skal serveres med xml data. Styring gjøres med bakgrunn i satte parametre
| fra klienten.
|*****/
require_once 'script_init.php';          # Setter opp rammer for appl.
require_once 'genXml.php';              # Sette opp xml for angitte kolonne.

/*Databasefeltene i tabell 'pr_produ' som det er aktuelt å hente data fra */
$dbFields=array(
  'ID', 'id_resal', 'id_brand', 'id_subca', 'produName', 'produUrl',
  'produDescript', 'price', 'currency', 'img_url', 'mod_nr', 'qey_word', 'dtg'
);

/* Lesing og tilordning av parameter i forespoersel fra klient */
if (!isset($_GET['col'])){              # Hvis kolonne ikke er satt så
  $params['col']=1;                     # skal kolonne default være 1.
} else {                                # Ellers så
  $params['col']=$_GET['col'];          # skal kolonne være som satt.
}
$col=$params['col'];

if (!isset($_GET['id'])){               # Hvis id ikke er satt så
  $params['id']='0';                    # skal id default være 0.
} else {                                # Ellers så
  $params['id']=$_GET['id'];            # skal id være som satt.
}
$id=$params['id'];

if (!isset($_GET['page'])){            # Hvis pan ikke er satt så
  $params['page']='0';                  # skal pan default være 0.
} else {                                # Ellers så
  $params['page']=$_GET['page'];        # skal pan være som satt.
}
$page=$params['page'];

if (!isset($_GET['alt'])){             # Hvis alt ikke er satt så
  $params['alt']='0';                  # skal alt default være 0.
} else {                                # Ellers så
  $params['alt']=$_GET['alt'];          # skal alt være som satt.
}
$alt=$params['alt'];

/* Tilordning av returdata til kolonne utfra forespoersel parameter 'col' */
header("Content-Type: text/xml; charset=UTF-8");#Sender XML header
openDb();
switch ($col):
  case $col==2: Col_2($id,$dbTable='pr_subca',$page,$alt);          # Kolonne 1
    break;
  case $col==3: Col_3($id,$dbTable='pr_produ',$page,$alt);          # Kolonne 2
    break;
  case $col==4: Col_4($id,$dbTable='pr_produ',$page,$alt);          # Kolonne 3
    break;
  case $col==5: Col_5($id,$dbFields,$dbTable='pr_produ',$alt);     # Kolonne 4
    break;
  default : Col_1($dbTable='pr_mainc',$alt);                          # Kolonne 5
endswitch;
closeDb();
?>

```



13. PHP kode for serversiden av prototyp. Skript: script_init.php

```

<?php
/*****\
| Scriptet styrer visning av feilmeldinger, inkluderer nødvendig mapper |
| og filer for kjøring av php scriptene, samt tekst bib. for applikasjon. |
\*****/

define ("DIR", 'webService');           # Angir rotmappe for applikasjon

/* Array med mappene som har nødvendig script for applikasjonen. ----- */
$dir=array(
    'include',                          # Applikasjonsuavhengige script.
    DIR.'/phpJoint',                   # Script som deles av hele php appl.
    DIR.'/phpXmlServ',                 # Script som genererer xml til klient.
    DIR.'/phpXmlAdm',                  # Script som vedlikeholder databasetab.
    DIR.'/fritekstsøk',                # Script som utfører tekstøk i prod DB.
);

/* Generell feilhaandtering i PHP ----- */
error_reporting(E_ALL);                # Rapporterer alle php-feil som oppstår.
ini_set("display_errors", 1);          # 1 :Viser feilmeldinger, 0 :Viser ikke.

/* Generering av data basert på $dir-array for å sette "include_path" --- */
$inclDir='.';
    foreach ($dir as $start => $end) {
        $inclDir.="../".$end.'/';      # Kode for windows server
    /* $inclDir.="../".$end.'/'; */    # Kode for linux server
    }
$inclDir=substr_replace($inclDir,'',-1,1);
ini_set("include_path", "$inclDir");

/* Innkludering av filer plassert i "include" og "phpJoint". ----- */
require_once 'apl_db_funk.php';        # Databasesperrings arkiv for appl.
require_once 'funk_database.php';     # Generelle database relaterte funk.
require_once 'script_DB_lang.php';    # Database tilkoblingsdata
require_once 'funk_html.php';         # Html 4.01 strickt genererings funk.
require_once 'script_lang.php';       # Språkarkiv med gen. ord og setninger.
require_once 'funk_other.php';        # Andre ikke-applikasjonsavhengige funk.

/* Tekst bibliotek for denne applikasjonens forskjellige script: ----- */
define('STR_MENY', "FORBRUKERTIPSET.NO ADM-MENY BETA 1:");
define('STR_PARSE', "1. Parse data fra xml fil inn i databasetabellen");
define('STR_SUCSESS', "Det funket - Tilbak til meny");
define('STR_UPD_START', "Tradedoubler oppdaterings - script startet :");
define('STR_UPD_ROWS', "Ant. produkter behandlet ifm oppdateringen :");
define('STR_UPD_ENDS', "Tradedoubler oppdaterings-script avsluttet :");
?>

```



14. PHP kode for serversiden av prototyp. Skript: genXml.php

```

<?php
/.....\
| Hjelpesfunksjoner for tilordning av returdata i de spesifikke kolonnene |
\.....\

/*Funksjonen genererer data i xml format til Interesseseft kolonnen (1) */
function Col_1($dbTable, $salt){
$xmlRoot='menu';           #Xml <root> tag
$xmlRootChild1='lev';     #Xml <rootchild> tag
$xmlRootChild2='alt';     #Xml <rootchild> tag
$xmlRootChild3='cat';     #Xml <rootchild> tag
$Attr='id';               #Xml <rootchild> tag attributt

$doc = new DOMDocument(); #Oppretter objektet $doc av klassen DOMDocument
$doc->formatOutput = true;

    $root=$doc->createElement($xmlRoot); #Oppretter rot element
    $doc->appendChild($root); #Initierer under elementer til rotelementet
    // Oppretter "child" element
    $childNode = $doc->createElement($xmlRootChild1);
    $root->appendChild($childNode);
    // Oppretter "tekst" node til overstaaende "child" element
    $childNodeValue = $doc->createTextNode(1);
    $childNode->appendChild($childNodeValue);

    $result=getDbData('col_1', $dbTable);
    $items=0;
    while ($row=fetchRowDb($result)) { #looper gjennom databasen rad for rad
        if($row['ID']!=3){#Filtrering av erotikk kategori.
            $items++;
            // Oppretter "child" element
            $childNode = $doc->createElement($xmlRootChild3);
            $root->appendChild($childNode);
            // Oppretter "tekst" node til overstaaende "child" element
            $childNodeValue = $doc->createTextNode($row['maincName']);
            $childNode->appendChild($childNodeValue);
            // Oppretter attributt node
            $childNodeAttr2 = $doc->createAttribute($Attr);
            $childNode->appendChild($childNodeAttr2);
            // Oppretter attributt verdi til overstaaende attributt
            $childNodeAttrValue2 = $doc->createTextNode($row['ID']);
            $childNodeAttr2->appendChild($childNodeAttrValue2);
        }
    }
    // Oppretter "child" element
    $childNode = $doc->createElement($xmlRootChild2);
    $root->appendChild($childNode);
    // Oppretter "tekst" node til overstaaende "child" element
    $childNodeValue = $doc->createTextNode($items);
    $childNode->appendChild($childNodeValue);
    $xmlStr = $doc->saveXML(); #Dumper xml data til en tekststreng
    print ($xmlStr);
}

/*Funksjonen genererer data i xml format til Hovedkategorier kolonnen (2) */
function Col_2($id, $dbTable, $salt){
$xmlRoot='menu';           #Xml <root> tag
$xmlRootChild1='lev';     #Xml <rootchild> tag
$xmlRootChild2='alt';     #Xml <rootchild> tag
$xmlRootChild3='cat';     #Xml <rootchild> tag
$Attr='id';               #Xml <rootchild> tag attributt

$doc = new DOMDocument(); #Oppretter objektet $doc av klassen DOMDocument

```



```

$doc->formatOutput = true;

$root=$doc->createElement($xmlRoot); #Oppretter rot element
$doc->appendChild($root); #initierer under elementer til rotelementet

// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild1);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode(2);
$childNode->appendChild($childNodeValue);

$result=getDbData('col_2', $dbTable, $id);
$items=0;
while ($row=fetchRowDb($result)) { #looper gjennom databasen rad for rad
    $items++;
    // Oppretter "child" element
    $childNode = $doc->createElement($xmlRootChild3);
    $root->appendChild($childNode);
    // Oppretter "tekst" node til overstaaende "child" element
    $childNodeValue = $doc->createTextNode($row['subcaName']);
    $childNode->appendChild($childNodeValue);
    // Oppretter attributt node
    $childNodeAttr2 = $doc->createAttribute($Attr);
    $childNode->appendChild($childNodeAttr2);
    // Oppretter attributt verdi til overstaaende attributt
    $childNodeAttrValue2 = $doc->createTextNode($row['ID']);
    $childNodeAttr2->appendChild($childNodeAttrValue2);
}

// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild2);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($items);
$childNode->appendChild($childNodeValue);
$xmlStr = $doc->saveXML(); #Dumper xml data til en tekststreng
print ($xmlStr);
}
/*Funksjonen genererer data i xml format til Prisfiltrerings kolonnen (3) */
function Col_3($id,$dbFields,$dbTable,$alt){
$xmlRoot='menu';           #Xml <root> tag
$xmlRootChild1='lev';      #Xml <rootchild> tag
$xmlRootChild2='alt';      #Xml <rootchild> tag
$xmlRootChild3='cat';      #Xml <rootchild> tag
$Attr='id';                #Xml <rootchild> tag attributt

$doc = new DOMDocument(); #Oppretter objektet $doc av klassen DOMDocument
$doc->formatOutput = true;

$root=$doc->createElement($xmlRoot); #Oppretter rot element
$doc->appendChild($root); #Initierer under elementer til rotelementet

// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild1);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode(3);
$childNode->appendChild($childNodeValue);

$items=0;
foreach (priceCat() as $beg => $end) {

```



```

if(getDbData('col_3','pr_produ',$id,$beg,$end)){
    $items++;
    // Oppretter "child" element
    $childNode = $doc->createElement($xmlRootChild3);
    $root->appendChild($childNode);
    // Oppretter "tekst" node til overstaaende "child" element
    $childNodeValue = $doc->createTextNode($end);
    $childNode->appendChild($childNodeValue);
    // Oppretter attributt node
    $childNodeAttr2 = $doc->createAttribute($Attr);
    $childNode->appendChild($childNodeAttr2);
    // Oppretter attributt verdi til overstaaende attributt
    $childNodeAttrValue2 = $doc->createTextNode($beg);
    $childNodeAttr2->appendChild($childNodeAttrValue2);
}
}
// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild2);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($id);
$childNode->appendChild($childNodeValue);
$xmlStr = $doc->saveXML(); #Dumper xml data til en tekststreng
print($xmlStr);
}
/*Funksjonen genererer data i xml format til Produkter kolonnen (4) */
function Col_4($id,$dbFields,$dbTable,$salt){
$xmlRoot='menu';           #Xml <root> tag
$xmlRootChild1='lev';     #Xml <rootchild> tag
$xmlRootChild2='alt';     #Xml <rootchild> tag
$xmlRootChild3='cat';     #Xml <rootchild> tag
$Attr='id';               #Xml <rootchild> tag attributt

$doc = new DOMDocument(); #Oppretter objektet $doc av klassen DOMDocument
$doc->formatOutput = true;

$root=$doc->createElement($xmlRoot); #Oppretter rot element
$doc->appendChild($root); #Initierer under elementer til rotelementet

// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild1);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode(4);
$childNode->appendChild($childNodeValue);

foreach (priceCat() as $beg => $end) {
    if ($beg==$id)
        $result=getDbData('col_4', $dbTable,$salt,$beg,$end);
}
}
$items=0;
while ($row=fetchRowDb($result)) { #looper gjennom databasen rad for rad
    $items++;
    // Oppretter "child" element
    $childNode = $doc->createElement($xmlRootChild3);
    $root->appendChild($childNode);
    // Oppretter "tekst" node til overstaaende "child" element
    $childNodeValue = $doc->createTextNode($row['produName']);
    $childNode->appendChild($childNodeValue);
    // Oppretter attributt node
    $childNodeAttr2 = $doc->createAttribute($Attr);
    $childNode->appendChild($childNodeAttr2);
}
}

```



```

// Oppretter attributt verdi til overstaaende attributt
$childNodeAttrValue2 = $doc->createTextNode($row['ID']);
$childNodeAttr2->appendChild($childNodeAttrValue2);
}
// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild2);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($items);
$childNode->appendChild($childNodeValue);
$xmlStr = $doc->saveXML(); #Dumper xml data til en tekststreng
print($xmlStr);
}
/*Funksjonen genererer data i xml format til Poduktdetalj kolonnen (5) */
function Col_5($id,$dbFields,$dbTable,$alt){
$xmlRoot='product'; #Xml <root> tag
$xmlRootChild1='lev'; #Xml <rootchild> tag
$xmlRootChild2='alt'; #Xml <rootchild> tag
$Attr='id'; #Xml <rootchild> tag attributt

$doc = new DOMDocument(); #Oppretter objektet $doc av klassen DOMDocument
$doc->formatOutput = true;

$root=$doc->createElement($xmlRoot); #Oppretter rot element
$doc->appendChild($root); #Initierer under elementer til rotelementet

// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild1);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode(5);
$childNode->appendChild($childNodeValue);

$result=getDbData('col_5', $dbTable, $id);
while ($row=fetchRowDb($result)) { #looper gjennom databasen rad for rad
// Oppretter "child" element
$childNode = $doc->createElement('merchant');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($row[$dbFields[1]]);
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('name');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($row[$dbFields[4]]);
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('description');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($row[$dbFields[6]]);
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('image_url');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($row[$dbFields[9]]);
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('buy_url');
$root->appendChild($childNode);
}

```



```

// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($row[$dbFields[5]]);
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('price');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($row[$dbFields[7]]);
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('search_name');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode($row[$dbFields[11]]);
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('category');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode('tom');
$childNode->appendChild($childNodeValue);
// Oppretter "child" element
$childNode = $doc->createElement('brand');
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode('tom');
$childNode->appendChild($childNodeValue);
}
// Oppretter "child" element
$childNode = $doc->createElement($xmlRootChild2);
$root->appendChild($childNode);
// Oppretter "tekst" node til overstaaende "child" element
$childNodeValue = $doc->createTextNode(1);
$childNode->appendChild($childNodeValue);
$xmlStr = $doc->saveXML(); #Dumper xml data til en tekststreng
print($xmlStr);
}
/*Funksjonen genererer priskategori array som brukes i kolonne 3 og 4 */
function priceCat(){
$pricCat=array(
    1      =>100,
    100   =>300,
    300   =>500,
    500   =>750,
    750   =>2000,
    2000  =>4000,
    4000  =>6000,
    6000  =>8000,
    8000  =>10000,
    10000 =>12000,
    12000 =>14000,
    14000 =>18000,
    18000 =>22000,
    22000 =>30000,
    30000 =>99000,
);
return $pricCat;
}
?>

```

15. PHP kode for serversiden av prototyp. Skript: apl_db_funk.php

```

<?php
/*****
| Skriptet ivaretar nødvendige spørringer mot databasetabellene ved
| generering av nye id'er og oppdatering av eksisterende.
| Skriptet haandterer også nødvendige spørringer ifm sortering/soek av data
| til klient. Det benyttes databasefunksjoner fra funk_database.php fila.
| *****/
/*Funksjonen utfører dataspørringer avhengig av input parametre */
function getDbData($query,$table,$inp1='', $inp2='', $inp3='', $inp4='', $inp5='') (
$queryStr = array(
#Generelle databasespørringer ----->
'exist' => "select COUNT(*) from $table where $inp1=$inp2",
'nrRow' => "select COUNT(*) from $table",
'allIn' => "select * from $table",
#genXml.php spesefikke databasespørringer ----->
'col_1' => "select * from $table where id>1",
'col_2' => "select * from $table where id_mainc='$inp1' and ID>1 limit $inp2,
$inp3",
'all_2' => "select COUNT(*) from $table where id_mainc='$inp1' and ID>1",
'col_3' => "select distinct id_prica from $table where id_subca='$inp1'
order by id_prica asc",
'getPr' => "select stop from $table where start='$inp1'",
'col_4' => "select * from $table where id_subca='$inp1'
and price>='$inp2' and price<='$inp3' limit $inp4, $inp5",
'all_4' => "select COUNT(*) from $table where id_subca='$inp1'
and price>='$inp2' and price<='$inp3'",
'col_5' => "select * from $table where id=$inp1",
#readToProdTab.php spesefikke databasespørringer ----->
'split' => "select * from $table limit $inp1, $inp2",
'price' => "select * from $table where price='$inp1'",
'd_t_g' => "select UNIX_TIMESTAMP(dtg) from $table where id='$inp1'",
'oneId' => "select ID from $table where produName='$inp1'",
'proCa' => "select subcat from $table where id='$inp1'",
'priCa' => "select start from $table where start<='$inp1' and stop> '$inp1'",
'getRe' => "select ID from $table where resalName='$inp1'",
'getBr' => "select ID from $table where brandName='$inp1'",
#updateIdFeeds.php spesefikke databasespørringer ----->
'dropT' => "drop table if exists $table",
'copyC' => "INSERT INTO $table SELECT * FROM $inp1",
'renam' => "RENAME TABLE $table TO $inp1",
'prodF' => "select ID, id_subca, produName, produDescript, qey_word from
$table",
'subCa' => "select subcaName from $table where id='$inp1'",
);
if ($query=='exist' || $query=='nrRow' || $query=='getPr' || $query=='all_2'
|| $query=='all_4' || $query=='price' || $query=='d_t_g' || $query=='oneId'
|| $query=='proCa' || $query=='priCa' || $query=='iniId' || $query=='subCa') {
$result=queryDb($queryStr[$query]);
return fetchValueDb($result);
}else {
return $result=queryDb($queryStr[$query]);
}
}
/*Funksjonen skriver nytt produkt til databasetab 'pr_produ'. */
function newProd($dbTab,$idResal,$idBrand,$idSubCa,$idPriCa,$name,$pUrl,$desc,
$pric,$curr,$iUrl,$modN,$resa){
queryDb("INSERT INTO $dbTab SET id_resal='$idResal',id_brand='$idBrand',
id_subca='$idSubCa',id_prica='$idPriCa',produName='$name',produUrl='$pUrl
produDescript='$desc',price='$pric',currency='$curr',img_url='$iUrl
mod_nr='$modN',qey_word='$resa'");
}
?>

```




16. XHTML kode for prototypen. Index.html

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <?xml-stylesheet type="text/xsl" href="copy.xsl"?>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
4     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head>
7 <meta name="description" content="forbrukertipset er en gratis søke og sammenlign tjeneste som gir brukere produkter
8 fra flere hundre kategorier" />
9 <meta name="keywords" content="søk, produkt,sammenlign, navigasjon, anbefalt, best, pris" />
10 |
11 <title>Forbrukertipset - Søk og sammenlign beste pris på produkter</title>
12 <meta name="verify-v1" content="pHg7UMFD5fpkRjHAKBp4nBYtzbaTa3b5xbnek7rpxbM" />
13 <meta name="robots" content="index, follow" />
14
15 <meta http-equiv="content-language" content="no" />
16
17 <link type="text/css" rel="stylesheet" href="css/hero.css" media="screen"/>
18
19 <script type="text/javascript" src="js/newEngine.js"></script>
20 <script type="text/javascript" src="js/sweetLib.js"></script>
21
22 </head>
23 <body onload="first();">
24
25 <div id="topbox"></div>
26 <div id="logo"><a href="index.html">Forbrukertipset.no</a></div>
27
28 <div id="content" class="navsite"><!-- Main wrapper -->
29
30
31 <div id="introbox"><p>Velg fra listen og blå enkelt i mer enn 300 000 produkter.</p>
32 </div><!-- Intro-box end-->
33
34 <div id="browse">
35
36 <table border="0">
37 <tbody>
38 <tr id="contentRow">
39
40 <td id="backnav" title="Klikk for å komme tilbake" style="display:none" onclick="backNav();return false;">
41 <a class="level" href="#">'</a>
42 </td>
43
44 <!--Første Kolonne (Interesseparadde)-->
45 <td class="level" id="level-1">
46 <h3 class="series" id="series-1"></h3>
47 <div class="pane">
48 <div class="browse-total" id="total-1"></div>
49 <div class="paginate" id="navlinks1"></div>
50
51 <ul id="one">
52 <li><!-- Her fylles hovedkategorier --></li>
53 </ul>
54 </div>
55 </td>
56
57 <!--Andre Kolonne (Kategori)-->
58 <td class="level" id="level-2">
59 <h3 class="series" id="series-2"></h3>
60 <div class="pane">
61 <div class="browse-total" id="total-2"></div>
62 <div class="paginate" id="navlinks2"></div>
63
64 <ul id="two">
65 <li><!--Her fylles Subkategorier--></li>
66 </ul>
67
68 </div>
69 </td>
70
71 <!--Tredje Kolonne (Produkt)-->
72 <td class="level" id="level-3">
73 <h3 class="series" id="series-3"></h3>
74 <div class="pane">
75 <div class="browse-total" id="total-3"></div>
76 <div class="paginate" id="navlinks3"></div>
77
78 <ul id="tree">
79 <li><!--Her fylles Produkter--></li>
80 </ul>
81
82 </div>
83 </td>
84
85 <!--Fjerde Kolonne (Produkter)-->
86 <td class="level" id="level-4">
87 <h3 class="series" id="series-4"></h3>
88 <div class="pane">

```



```

89 <div class="browse-total" id="total-4"></div>
90 <div class="paginate" id="navlinks4"></div>
91
92 <ul id="four">
93 <li><!--Her kommer produktdetaljer--></li>
94 </ul>
95
96 </div>
97 </td>
98
99 <!--Femte Kolonne (Handlekurv)-->
100 <td class="level" id="level-5" style="display:none">
101 <h3 class="series" id="series-5">Detaljer</h3>
102 <div class="pane">
103 <div class="browse-total" id="total-5"></div>
104 <div class="paginate" id="navlinks5"></div>
105
106 <ul id="five">
107 <li><!--Her kommer produktdetaljer--></li>
108 </ul>
109
110 </div>
111 </td>
112
113 </tr>
114 </tbody>
115 </table>
116
117
118 </div><!--Browse Slutt-->
119
120 <!-- </div></div> -->
121 <!--Top left/right slutt-->
122
123 <div id="bottom-box">
124 <div id="support-arrow" class="column-1"></div>
125 <form id="searchBarForm" class="general" method="get" action="/" name="searchBarForm">
126 <label id="heading" for="searchFieldQuery"> Søk på hva du vil! (Bransje, firma, produkt, merke eller pris) </label>
127 <div class="fields">
128 <input id="searchFieldQuery" class="text w431" type="text" value="" name="Query"/>
129 <input id="queryType" type="hidden" value="8" name="qt"/>
130 </div>
131 <div class="actions">
132 <input class="button" type="submit" value="søk"/>
133 </div>
134 </form>
135
136 <div class="reset"></div>
137 </div><!--bottom Box slutt-->
138
139 </div> <!--content Slutt-->
140
141 <script type="text/javascript">
142 var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");
143 document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));
144 </script>
145 <script type="text/javascript">
146 var pageTracker = _gat._getTracker("UA-4111141-1");
147 pageTracker._initData();
148 pageTracker._trackPageview();
149 </script>
150 </body>
151 </html>

```



17. CSS kildekode for prototype. Hero.css

```

1 a:link {color: #455560;}
2 a:visited {color: #455560; }
3 a:hover {text-decoration: none; }
4 a img { border: 0;}
5 .red {color: #CC0000;}
6 p {margin: 0 0 1em;}
7
8 /*
9 -----
10 NAVIGATOR INTRO
11 -----*/
12 #introbox {
13     background: #EBD77E url(../images/navintro-nope.gif) no-repeat left center;
14     width: 907px;
15     border: 1px solid #B49E44;
16     border-width: 1px 1px 1px;
17     margin: 0px 0px -1px 0px;
18     min-height: 15px;
19     padding: 0px;
20     position: relative;}
21
22 #introbox p {
23     padding: 0px 5px 0px 100px;}
24 /*
25 -----
26 NAVIGATOR BOXES
27 -----*/
28 #box {
29     width: 910px;
30 }
31
32 #box-1 {
33     width: 226px;
34     min-height: 495px;
35     background-color: #fff;
36     border-left: 1px solid #b3b3b3;
37     border-top: 1px solid #601166;
38     float: left;
39 }
40
41 #box-2 {
42     width: 226px;
43     min-height: 495px;
44     background-color: #fff;
45     border-left: 1px solid #b3b3b3;
46     border-top: 1px solid #601166;
47     float: left;}
48
49 #box-3 {
50     width: 226px;
51     min-height: 495px;
52     background-color: #fff;
53     border-left: 1px solid #b3b3b3;
54     border-top: 1px solid #601166;
55     float: left;}
56
57 #backnav {
58     width: 226px;
59     min-height: 495px;
60     background-color: #fff;
61     border-left: 1px solid #b3b3b3;
62     border-top: 1px solid #601166;
63     float: left;}
64
65 #box-4 {
66     width: 226px;
67     min-height: 495px;
68     background-color: #fff;
69     border-left: 1px solid #b3b3b3;
70     border-right: 1px solid #b3b3b3;
71     border-top: 1px solid #601166;
72     float: left;}
73
74 #box-5 {
75     width: 452px;
76     min-height: 495px;
77     background-color: #fff;
78     border-left: 1px solid #b3b3b3;
79     border-right: 1px solid #b3b3b3;
80     border-top: 1px solid #601166;
81     float: left;}
82
83 #box h3 {
84     margin: 0;
85     padding: 6px 8px 4px 18px;
86     font-size: 110%;
87     color: #222;
88     border-bottom: 1px solid #b3b3b3;

```



```

89 /* background: url(../images/h3-boxes.gif) no-repeat top left; */
90 background-color: #e8e8e8;
91 }
92
93 /*
94 -----
95 NAVIGATOR UL MENU
96 -----*/
97 #box ul {
98     list-style: none;
99     margin: 0;
100     padding: 6px 8px 4px 5px;}
101
102 #box li {
103     margin: 0px;
104     padding: 0;
105     position: relative; }
106
107 #box ul a.coll,a.hi,a.open,#box ul a.detail {
108     background: #fff url(../images/right.gif) no-repeat 97% 50%;
109     display: block;
110     position: relative;
111     line-height: 1.25em;
112     margin: 0 3px 0 0 ;
113     padding: 5px 35px 5px 1.1em;
114     text-indent: -1em;
115     text-decoration: none; }
116
117 #box ul a.hi {
118     background: #455560 url(../images/rightW.gif) no-repeat 97% 50%;
119     color: #fff;}
120
121 #box ul a.detail {
122     background: #455560 url(../images/detailsicon.gif) no-repeat 97% 50%;
123     color: #fff;}
124
125 #box ul a.coll:hover {background: #455560 url(../images/right.gif) no-repeat 97% 50%; color: #fff;}
126
127 a.hi:hover,a.open:hover {
128     background: #005fb5 url(../images/rightW.gif) no-repeat 97% 50%;
129     color: #fff;
130     cursor: pointer;}
131
132 #box ul a.open {background: #ffe9c6 url(../images/rightW.gif) no-repeat 97% 50%; color: #222;}
133 #box ul a.open:hover { background-color: #ffe9c6;}
134
135 /*
136 -----
137 SUPPORT ARROW
138 -----*/
139 #clearsides {
140     clear: both;}
141
142 #bottom-box {
143     background: #455560;
144     width: 887px;
145     border: 1px solid #b3b3b3;
146     border-width: 1px 1px 1px;
147     margin: 0px 0px 5px 0px;
148     min-height: 10px;
149     padding: 10px;
150     position: relative;}
151
152
153 #support-arrow {
154     background: url(../images/support-arrowA.gif) no-repeat 0 0;
155     height: 21px;
156     position: absolute;
157     top: -20px;
158     width: 42px;}
159
160 .column-1 { left: 95px;}
161 .column-2 { left: 322px;}
162 .column-3 { left: 549px;}
163 .column-4 { left: 776px;}
164 .column-5 { left: 776px;}
165
166 /*
167 -----
168 NAVIGATOR HELPERS
169 -----*/
170 #box li#selecta {
171     height: 50px;
172     color: #555;
173     font-size: 150%;
174     font-weight: bold;
175     line-height: 1em;
176     margin: 10px auto 0;

```



```
177 padding: 20px 0 0 ;
178 position:static;
179 text-align: center;}
180
181 #box li#basket {
182 height: 140px;
183 background: url(../images/backbutton.gif) no-repeat center top; */
184 color: #555;
185 font-size: 159%;
186 font-weight: bold;
187 line-height: 1em;
188 margin: 5px auto 0;
189 padding: 30px 0 0 ;
190 position:static;
191 text-align: center; }
192
193 #box li#loading {
194 height: 144px;
195 text-align: center;
196 padding: 30px 0 0 ;
197 color: #555;
198 font-size: 115%;
199 font-weight: normal;
200 margin: 50px auto 0;
201 line-height: 1em;
202 list-style:none;}
203
204 /*
205 -----
206 NAVIGATOR PRODUCTDETAILS
207 -----*/
208 #previewImage .zoom {background: url(../images/zoom.png) no-repeat left center;
209 min-height: 10px;
210 padding: 2px 2px 2px 20px;}
211
212 #previewImage {
213 margin: 5px 10px;
214 border-top: double;
215 border-color: #b49e44;}
216
217 #previewImage a {
218 display: block;position: relative;min-height: 30px;}
219
220 #previewImage a img {display: block;
```



18. Kildekode JavaScript funksjonsbibliotek. sweetLib.js

```

1 |var IE = true
2
3 |/*-----
4 |NYTT XHR Cross-Browser Object lagt til 16.04.2008
5 |-----*/
6 |function sendRequest(url,callback,postData) {
7 |    var req = createXMLHTTPObject();
8 |    if (!req) return;
9 |    var method = (postData) ? "POST" : "GET";
10 |    req.open(method,url,true);
11 |    req.setRequestHeader('User-Agent','XMLHTTP/1.0');
12 |    if (postData)
13 |        req.setRequestHeader('Content-type','application/x-www-form-urlencoded');
14 |    req.onreadystatechange = function () {
15 |        if (req.readyState != 4) return;
16 |        if (req.status != 200 && req.status != 304) {
17 |            // alert('HTTP error ' + req.status);
18 |            return;
19 |        }
20 |        callback(req);
21 |    }
22 |    if (req.readyState == 4) return;
23 |    req.send(postData);
24 |}
25
26 |var XMLHttpFactories = [
27 |    function () {return new XMLHttpRequest();},
28 |    function () {return new ActiveXObject("Msxml2.XMLHTTP");},
29 |    function () {return new ActiveXObject("Msxml3.XMLHTTP");},
30 |    function () {return new ActiveXObject("Microsoft.XMLHTTP");}
31 |];
32
33 |function createXMLHTTPObject() {
34 |    var xmlhttp = false;
35 |    for (var i=0;i<XMLHttpFactories.length;i++) {
36 |        try {
37 |            xmlhttp = XMLHttpFactories[i]();
38 |        }
39 |        catch (e) {
40 |            continue;
41 |        }
42 |        break;
43 |    }
44 |    return xmlhttp;
45 |}
46
47 |/*-----
48 |Create element and create anchor
49 |-----*/
50
51 |function createElem(tag, content, className) {
52 |    elem=document.createElement(tag);
53 |    if (content!=null) {
54 |        if (typeof content=="string")
55 |            elem.appendChild(document.createTextNode(content));
56 |        else if (typeof content=="object")
57 |            elem.appendChild(content);
58 |    }
59 |    if (className!=null)
60 |        elem.setAttribute("class", className);
61 |    return elem;
62 |}
63
64 |function createA(content, href, title, className) {
65 |    elem=createElem("a", content, className);
66 |    elem.setAttribute("href", href);
67 |    if (href!=null)
68 |        elem.setAttribute("href", href);
69 |    if (href!=null)
70 |        elem.setAttribute("class", "coll");
71 |    if (title!=null)
72 |        elem.setAttribute("title", title);
73 |    return elem;
74 |}
75 |}

```



19. Kildekode JavaScript fil. newEngine.js

```

1  /*----- Global Variables -----*/
2  var maxColumns    = 4;
3  var currentColumn = 1;
4  var position;
5  var str1          = ("Velg");
6  var str2          = ("Interesse");
7
8  var str3          = ("Maks Pris NOK:");
9
10 var bodyElem;
11 var bodyElem2;
12 var bodyElem3;
13 var bodyElem4;
14 var menuElem;
15 var menuElem2;
16 var menuElem3;
17 var menuElem4;
18 var menuRes;
19 var lastSelected;
20 var lastSelected2;
21 var lastSelected3;
22 var lastSelected4;
23 var query;
24
25 /*----- Call Server for column 1-5 -----*/
26 function first() {
27     var level= ("1");
28     var position = ("two");
29     displayArrow(position, str1, str2);
30     displayLoading(level);
31     sendRequest("webService/xmlOutput.php", genFirstMenu);
32     bodyElem=document.getElementById("level-1");
33     menuElem=document.getElementById("one");
34     bodyElem.insertBefore(menuElem,menuElem);
35 }
36
37 function second(query) {
38     var position = ("tree");
39     var str2 = ("hovedkategori");
40     displayArrow(position, str1, str2);
41     sendRequest("webService/xmlOutput.php?"+query, genSecondMenu);
42     bodyElem2 = document.getElementById("level-2");
43     menuElem2 = document.getElementById("two");
44     bodyElem2.insertBefore(menuElem2,menuElem2);
45 }
46
47 function third(query) {
48     var position = ("four");
49     var str2 = ("kategori");
50     displayArrow(position, str1, str2);
51     sendRequest("webService/xmlOutput.php?"+query, genThirdMenu);
52     bodyElem3 = document.getElementById("level-3");
53     menuElem3 = document.getElementById("tree");
54     bodyElem3.insertBefore(menuElem3,menuElem3);
55 }
56
57 function fourth(query) {
58     sendRequest("webService/xmlOutput.php?"+query, genFourthMenu);
59     bodyElem4 = document.getElementById("level-4");
60     menuElem4 = document.getElementById("four");
61     bodyElem4.insertBefore(menuElem4,menuElem4);
62 }
63
64 function fifth(query) {
65     sendRequest("webService/xmlOutput.php?"+query, genFifthMenu);
66     bodyElem4 = document.getElementById("level-5");
67     menuElem4 = document.getElementById("five");
68     bodyElem4.insertBefore(menuElem4,menuElem4);
69 }
70
71 /*----- Generate Menus for column 1-5 -----*/
72 function genFirstMenu(req) {
73     var interests, i, name, column, id, alt, intElem, intItemElem;
74     if (req.readyState==4) {
75         menuRes = req.responseXML;
76         interests = menuRes.getElementsByTagName("cat");
77         for (i = 0; i<interests.length; i++) {
78             interest= interests.item(i);
79             name = interest.firstChild.nodeValue;
80             column = ("2");
81             alts = menuRes.getElementsByTagName("alt")[0];
82             alt = alts.firstChild.nodeValue;
83             id = interest.getAttribute("id");
84             intElem = createA(name, "#", "Velg "+name, "interest");
85             intElem.setAttribute("onclick", "selectCat(this, '"+column+"', '"+id+"', '"+alt+"')");
86             if (i==0) {
87                 lastSelected = intElem;
88                 ul = document.getElementById("level-1").getElementsByTagName("ul")[0];

```



```

89         clearList(ul);
90     }
91     intItemElem = createElem("li", intElem);
92     menuElem.appendChild(intItemElem);
93 }
94     var h3 = document.getElementById("series-1").innerHTML="";
95     var h3 = document.getElementById("series-1");
96     var str= document.createTextNode("Interessefelt");
97     h3.appendChild(str);
98 }
99 }
100
101 function genSecondMenu(req, query) {
102     var interests, i, name, column, id, alt, intElem, intItemElem;
103     if (req.readyState==4) {
104         menuRes = req.responseXML;
105         interests = menuRes.getElementsByTagName("cat");
106         for (i = 0; i<interests.length; i++) {
107             interest= interests.item(i);
108             name = interest.firstChild.nodeValue;
109             column = ("3");
110             alts = menuRes.getElementsByTagName("alt")[0];
111             alt = alts.firstChild.nodeValue;
112             id = interest.getAttribute("id");
113             intElem = createA(name, "#", "Velg "+name, "interest");
114             intElem.setAttribute("onclick", "selectSubCat(this, '"+column+"', '"+id+"', '"+alt+"')");
115             if (i==0) {
116                 lastSelected2 = intElem;
117                 ul = document.getElementById("level-2").getElementsByTagName("ul")[0];
118                 clearList(ul);
119             }
120             intItemElem = createElem("li", intElem);
121             menuElem2.appendChild(intItemElem);
122         }
123         var h3 = document.getElementById("series-2").innerHTML="";
124         var h3 = document.getElementById("series-2");
125         var str= document.createTextNode("Hovedkategorier");
126         h3.appendChild(str);
127     }
128 }
129
130 function genThirdMenu(req, query) {
131     var interests, i, name, column, id, alt, intElem, intItemElem;
132     if (req.readyState==4) {
133         menuRes = req.responseXML;
134         interests = menuRes.getElementsByTagName("cat");
135         for (i = 0; i<interests.length; i++) {
136             interest= interests.item(i);
137             name = interest.firstChild.nodeValue;
138             column = ("4");
139             alts = menuRes.getElementsByTagName("alt")[0];
140             alt = alts.firstChild.nodeValue;
141             id = interest.getAttribute("id");
142             intElem = createA(str3 + name, "#", "Velg "+name, "interest");
143             intElem.setAttribute("onclick", "selectProd(this, '"+column+"', '"+id+"', '"+alt+"')");
144             if (i==0) {
145                 lastSelected3 = intElem;
146                 ul = document.getElementById("level-3").getElementsByTagName("ul")[0];
147                 clearList(ul);
148             }
149             intItemElem = createElem("li", intElem);
150             menuElem3.appendChild(intItemElem);
151         }
152         var h3 = document.getElementById("series-3").innerHTML="";
153         var h3 = document.getElementById("series-3");
154         var str= document.createTextNode("Kategorier");
155         h3.appendChild(str);
156     }
157 }
158
159 function genFourthMenu(req, query) {
160     var interests, i, name, column, id, alt, intElem, intItemElem;
161     if (req.readyState==4) {
162         menuRes = req.responseXML;
163         interests = menuRes.getElementsByTagName("cat");
164         for (i = 0; i<interests.length; i++) {
165             interest= interests.item(i);
166             name = interest.firstChild.nodeValue;
167             column = ("5");
168             alts = menuRes.getElementsByTagName("alt")[0];
169             alt = alts.firstChild.nodeValue;
170             id = interest.getAttribute("id");
171             intElem = createA(name, "#", "Velg "+name, "interest");
172             intElem.setAttribute("onclick", "selectDetails(this, '"+column+"', '"+id+"', '"+alt+"')");
173             if (i==0) {
174                 lastSelected4 = intElem;
175                 ul = document.getElementById("level-4").getElementsByTagName("ul")[0];
176                 clearList(ul);

```




```

177     }
178     intItemElem = createElem("li", intElem);
179     menuElem4.appendChild(intItemElem);
180 }
181     var h3 = document.getElementById("series-4").innerHTML="";
182     var h3 = document.getElementById("series-4");
183     var str= document.createTextNode("Produkter");
184     h3.appendChild(str);
185 }
186 }
187
188 function genFifthMenu(req, query) {
189     var details, i, name, column, id, alt, intElem, intItemElem;
190     if (req.readyState==4) {
191         menuRes = req.responseXML;
192         product = menuRes.getElementsByTagName("name")[0]
193         productName = product.childNodes[0];
194         price = menuRes.getElementsByTagName("price")[0]
195         priceAmount = price.childNodes[0];
196         imageu = menuRes.getElementsByTagName("image_url")[0]
197         imageUrl = imageu.childNodes[0];
198         desc = menuRes.getElementsByTagName("description")[0]
199         description = desc.childNodes[0];
200         buyu = menuRes.getElementsByTagName("buy_url")[0]
201         buy_url = buyu.childNodes[0];
202         merchant = menuRes.getElementsByTagName("merchant")[0]
203         merchantName= merchant.childNodes[0];
204         brand = menuRes.getElementsByTagName("brand")[0]
205         brandName = brand.childNodes[0];
206
207
208         a = (productName.nodeValue);
209         a1 = createElem("li", a);
210         b = (priceAmount.nodeValue);
211         b2 = createElem("li", b);
212         c = (imageUrl.nodeValue);
213         c3 = createElem("li", c);
214         d = (description.nodeValue);
215         d4 = createElem("li", d);
216         e = (buy_url.nodeValue);
217         e5 = createElem("li", e);
218         f = (merchantName.nodeValue);
219         f6 = createElem("li", f);
220         g = (brandName.nodeValue);
221         g7 = createElem("li", g);
222
223
224         ul = document.getElementById("level-5").getElementsByTagName("ul")[0];
225         clearList(ul);
226
227         var span = document.createElement('span');
228         span.setAttribute("class", "preview_title");
229         var str1 = document.createTextNode(a);
230         span.appendChild(str1);
231
232         var stars = document.createElement("img");
233         stars.setAttribute("src", "images/stars.gif");
234         stars.setAttribute("alt", "Popul r");
235
236
237         var div = document.createElement('div');
238         div.setAttribute("id", "previewImage");
239         div.appendChild(span);
240         div.appendChild(stars);
241
242         var a = document.createElement('a');
243         a.setAttribute("class", "zoom");
244         a.setAttribute("href", "#");
245         text = document.createTextNode(" Zoom bilde");
246         a.appendChild(text);
247
248         var image = document.createElement("img");
249         image.setAttribute("width", "180");
250         image.setAttribute("height", "180");
251         image.setAttribute("src", c);
252         image.setAttribute("alt", "ProduktBilde");
253         div.appendChild(image);
254         div.appendChild(a);
255         var li = document.createElement('li');
256         li.appendChild(div);
257         menuElem4.appendChild(li);
258
259         var para = document.createElement("h4");
260         var priceamount = document.createTextNode(b);
261         var pricetext = document.createTextNode("NOK ");
262         para.appendChild(pricetext);
263         para.appendChild(priceamount);
264         var li2 = document.createElement("li");

```



```

265     li2.appendChild(para);
266     menuElem4.appendChild(li2);
267
268     var para2      = document.createElement("p");
269     var deliverer  = document.createTextNode(f);
270     var delivererText = document.createTextNode("Leveret av: ");
271     para2.appendChild(delivererText);
272     para2.appendChild(deliverer);
273     var li3 = document.createElement("li");
274     li3.appendChild(para2);
275     menuElem4.appendChild(li3);
276
277
278     var image2 = document.createElement("img");
279     image2.setAttribute("src", "images/order.png");
280     image2.setAttribute("alt", "Bestill produkt");
281
282     var buyUrl = document.createElement("a");
283     buyUrl.setAttribute("href", e);
284
285     buyUrl.appendChild(image2);
286     li4 = document.createElement("li");
287     li4.appendChild(buyUrl);
288     menuElem4.appendChild(li4);
289
290     li5 = document.createElement("li");
291     var div2 = document.createElement('div');
292     div2.setAttribute("id", "bottom");
293     li5.appendChild(div2);
294     menuElem4.appendChild(li5);
295 }
296 }
297
298 /*----- Eventhandler for column 1-5 -----*/
299 function selectCat(elem, column, id, alt) {
300     var level= ("2");
301     displayLoading(level);
302
303     level3 = document.getElementById("level-3").getElementsByTagName("ul")[0];
304     clearList(level3);
305     level4 = document.getElementById("level-4").getElementsByTagName("ul")[0];
306     clearList(level4);
307     level5 = document.getElementById("level-5").getElementsByTagName("ul")[0];
308     clearList(level5);
309
310     lastSelected.setAttribute("class", "coll");
311     elem.setAttribute("class", "hi");
312     lastSelected = elem;
313     supportArrow();
314     var query = ("col="+column+"&"+"id="+id+"&"+"alt="+alt);
315     return !second(query);
316 }
317
318 function selectSubCat(elem, column, id, alt) {
319     var level= ("3");
320     displayLoading(level);
321
322     level4 = document.getElementById("level-4").getElementsByTagName("ul")[0];
323     clearList(level4);
324     level5 = document.getElementById("level-5").getElementsByTagName("ul")[0];
325     clearList(level5);
326
327     lastSelected.setAttribute("class", "open");
328     elem.setAttribute("class", "hi");
329     lastSelected = elem;
330     supportArrow2();
331     var query = ("col="+column+"&"+"id="+id+"&"+"alt="+alt);
332     return !third(query);
333 }
334
335 function selectProd(elem, column, id, alt) {
336     var level= ("4");
337     displayLoading(level);
338
339     level5 = document.getElementById("level-5").getElementsByTagName("ul")[0];
340     clearList(level5);
341
342     lastSelected.setAttribute("class", "open");
343     elem.setAttribute("class", "hi");
344     lastSelected = elem;
345     supportArrow3();
346     var query = ("col="+column+"&"+"id="+id+"&"+"alt="+alt);
347     return !fourth(query);
348 }
349
350 function selectDetails(elem, column, id, alt) {
351     var level= ("5");
352     displayLoading(level);

```



```

353 lastSelected.setAttribute("class", "open");
354 elem.setAttribute("class", "detail");
355 lastSelected = elem;
356 supportArrow3();
357 showHide();
358 var query = ("col="+column+"&"+"id="+id+"&"+"alt="+alt);
359 return !fifth(query);
360 }
361
362 /*----- Show bottomArrow for position-----*/
363 function supportArrow() {
364     arrow1 = document.getElementById("support-arrow");
365     arrow1.setAttribute("class", "column-2");
366 }
367
368 function supportArrow2() {
369     arrow1 = document.getElementById("support-arrow");
370     arrow1.setAttribute("class", "column-3");
371 }
372
373 function supportArrow3() {
374     arrow1 = document.getElementById("support-arrow");
375     arrow1.setAttribute("class", "column-4");
376 }
377
378
379 /*----- Display loading message -----*/
380 function displayLoading(level) {
381     if(document.getElementById("level-"+level)) {
382         ul = document.getElementById("level-" + level).getElementsByTagName('ul')[0];
383         clearList(ul);
384         var str1 = document.createTextNode("laster");
385         var li = document.createElement('li');
386         var img = document.createElement("img");
387         img.setAttribute("src", "images/process.gif");
388         li.setAttribute("id", "loading");
389         li.appendChild(img);
390         li.appendChild(str1);
391         ul.appendChild(li);
392     }
393 }
394
395 /*----- Display tips indicator -----*/
396 function displayArrow(position, str1, str2) {
397     var ul = document.getElementById(position);
398     var li = document.createElement("li");
399     li.setAttribute("id", "selecta");
400     var img = document.createElement("img");
401     img.setAttribute("src", "images/arrowLeft.gif");
402     var br = document.createElement("br");
403     var str1 = document.createTextNode(str1);
404     var br2 = document.createElement("br");
405     var str2 = document.createTextNode(str2);
406     li.appendChild(img);
407     li.appendChild(br);
408     li.appendChild(str1);
409     li.appendChild(br2);
410     li.appendChild(str2);
411     ul.appendChild(li);
412 }
413
414
415 /*----- Removers -----*/
416 function clearList(target) {
417     while (target.childNodes.length > 0) {
418         target.removeChild(target.childNodes[target.childNodes.length - 1]);
419     }
420 }
421
422 function removeArrow() {
423     document.getElementById("selecta").style.display = "none";
424 }
425
426 function removeBasket() {
427     document.getElementById("basket").innerHTML = "";
428 }
429
430 /*----- Show or hide column 5 -----*/
431 function showHide() {
432     var level1 = document.getElementById("level-1").style.display="none";
433     var level5 = document.getElementById("level-5").style.display="";
434     var backnav = document.getElementById("backnav").style.display="";
435     var bottom = document.getElementById("bottom-box");
436     bottom.style.width = "922px";
437     var top = document.getElementById("introbox");
438     top.style.width = "942px";
439 }
440

```



```
441 /*---- display backlink based on show/hide ----*/
442 function backNav(elem) {
443   var backnav= document.getElementById("backnav").style.display="none";
444   var level5 = document.getElementById("level-5").style.display="none";
445   var level1 = document.getElementById("level-1").style.display="";
446   var bottom = document.getElementById("bottom-box");
447   bottom.style.width = "903px";
448   var top    = document.getElementById("top-box");
449   top.style.width = "923px";
450 }
451
```