



# WAP - tjeneste for sanntidskoordinering

Diplomoppgave ved sivilingeniørutdanningen  
i  
informasjons og kommunikasjonsteknologi,  
Høgskolen i Agder.

av  
Frode Solvang

Grimstad, Mai 2000

# Sammendrag

Oppgaven tar for seg teori rundt koordinering, sanntid og Wireless Application Protocol. Den teoretiske studien viser at det er mulig å lage en tjeneste for WAP som kan brukes som et verktøy for koordinering. Som eksempel tar oppgaven for seg temaet kjørelag / ”kompiskjøring” som innebærer at folk som skal samme vei kan haike med hverandre. Videre viser studien at det med dagens teknologi ikke lar seg gjøre å lage en tjeneste for koordinering i sanntid innenfor temaet kjørelag. En slik tjeneste krever bedre interaktivitet og meldingsutveksling i WAP teknologien enn det WAP standarden gir mulighet for i dag.

Det er realisert et eksempel på en tjeneste for koordinering av kjørelag som kan aksesseres både fra web og WAP. Tjenesten er skrevet i Active Server Pages teknologi (ASP) som gjør det mulig å generere dynamiske sider både for web og WAP.

Et kjørelag organisert rundt en wap tjeneste vil være dynamisk i organisasjon og lite basert på regler. For at kjørelaget skal opprettholdes må sjåføren få kompensasjon for sin kjøring, enten ved betaling eller i form av gjenytelser. Dersom gjenytelsene er i form av at man bytter på å kjøre vil organisasjonen dreie fra å være svært dynamisk til å bli mer statisk og regelstyrt.

# Forord

Denne diplomoppgaven er skrevet som avslutning på sivilingeniør utdanningen innen Informasjons- og Kommunikasjonsteknologi (IKT) ved Høgskolen i Agder, institutt for informasjonsteknologi i Grimstad. Diplom oppgaven tilsvarer ett semesters arbeid, 10 vekttall.

Jeg vil takke Telenor FoU ved Trond Friisø og Alf Martin Sollund som har kommet med ide og forslag til oppgaven. En stor takk til min veileder ved HiA, Rune Fensli, for gode råd og innspill i forkant og underveis i arbeidet med oppgaven.

Grimstad, Mai 2000

Frode Solvang

# Innholdsfortegnelse

<b>1 INNLEDNING</b> .....	<b>5</b>
1.1 OPPGAVETEKST.....	5
<b>2 PROBLEMBESKRIVELSE</b> .....	<b>6</b>
2.1 INNLEDNING.....	6
2.2 KOORDINERING.....	7
2.2.1 Avhengigheter.....	8
2.2.2 Virtuelle grupper.....	11
2.2.3 Nye aktører i markedet.....	13
2.3 KRAV TIL ET SANNTIDSSYSTEM ELLER EN SANNTIDSTJENESTE.....	14
2.4 HVILKE TYPER TJENESTER ER DET BEHOV FOR ?.....	15
2.5 WIRELESS APPLICATION PROTOCOL (WAP).....	16
2.5.1 Hva er WAP ?.....	16
2.5.2 Teknologien bak WAP.....	17
2.5.3 Wireless Markup Language (WML).....	18
2.5.4 WML-script.....	19
2.5.5 Designprinsipper for WAP applikasjoner.....	20
<b>3 METODE</b> .....	<b>22</b>
3.1 INNLEDNING.....	22
3.1.1 Krav til en Wap applikasjon.....	22
3.2 DAGENS SITUASJON.....	23
3.2.1 Sterk grad av regler.....	23
3.2.2 Delvis regelbasert.....	23
3.2.3 Liten grad av regler.....	23
3.2.4 Aktører.....	24
3.3 ANALYSE AV AVHENGIGHETSRELASJONENE.....	25
3.3.1 Avhengighetens balanse.....	25
3.3.2 Avhengighetens struktur.....	25
3.3.3 Avhengighetens eksistens.....	25
3.3.4 Avhengighetens klima.....	26
3.3.5 Avhengighetens tema.....	26
3.4 HYPOTESE.....	27
3.4.1 Innledning.....	27
3.4.2 Scenario.....	27
3.5 EN NY TJENESTE BLIR TIL.....	28
3.6 APPLIKASJONSLØSNING.....	30
3.6.1 Innledning.....	30
3.6.2 Web grensesnitt.....	30
3.6.3 WAP grensesnitt.....	30

<b>4</b>	<b>UTVIKLING</b>	<b>31</b>
4.1	UTVIKLINGSVERKTØY	31
4.2	PROGRAMMERING	32
4.2.1	Active Server Pages (ASP)	32
4.3	SELVE TJENESTEN	33
4.3.1	Databasestruktur	33
4.3.2	Applikasjonsstruktur – Web grensesnitt	34
4.3.3	Applikasjonsstruktur – WAP/WML grensesnitt	35
4.3.4	Flytskjema	37
<b>5</b>	<b>RESULTAT</b>	<b>38</b>
5.1	WAP TJENESTEN	39
<b>6</b>	<b>DRØFTING</b>	<b>42</b>
6.1	ET NYTT SCENARIO	43
<b>7</b>	<b>KONKLUSJON</b>	<b>44</b>
<b>8</b>	<b>REFERANSER</b>	<b>45</b>
<b>9</b>	<b>VEDLEGG</b>	<b>47</b>
9.1	WML/ASP FILER	47
9.1.1	Vedlegg 1 : start.asp	47
9.1.2	Vedlegg 2 : SearchQuery.asp	49
9.1.3	Vedlegg 3 : InputQuery.asp	55
9.1.4	Vedlegg 4 : wmlres.asp	56
9.2	HTML/ASP FILER	57
9.2.1	Vedlegg 5 : default.asp	57
9.2.2	Vedlegg 6 : bruker.asp	58
9.2.3	Vedlegg 7 : pass2.asp	59
9.2.4	Vedlegg 8 : reserver2.asp	65
9.2.5	Vedlegg 9 : Calendar2.asp	66
9.2.6	Vedlegg 10 : reserver.asp	73
9.2.7	Vedlegg 11 : mystyle.css	74
9.3	FELLES FILER	75
9.3.1	Vedlegg 12 : file.dsn	75

# 1 Innledning

## 1.1 Oppgavetekst

### **Wap-tjeneste for sanntidskoordinering.**

Wap er en teknologi som gjør det mulig i større grad enn tidligere å lage tjenester for sanntidskoordinering. Teknologien gir mulighet for å levere Internett tjenester på små håndholdte terminaler.

Ut i fra et teoretisk studium av relevante egenskaper og aspekter rundt WAP, skal en se på mulighetene for å etablere en tjeneste for sanntidskoordinering og ressursreservering. Det skal også vurderes hvilke aspekter rundt en slik tjeneste som er viktige for at tjenesten skal lykkes.

Videre skal det utvikles en applikasjon som realiserer en tjeneste for sanntidskoordinering. En slik tjeneste kan være koordinering av ”kompiskjøring”, der en deler på å kjøre til / fra jobb i Grimstad. Personer som er interessert kan melde seg på tjenesten, på samme måte som en deltar i en nyhetsgruppe på Internett. Det er viktig at det lages et enkelt grensesnitt, beregnet for mobile enheter.

## **2 Problembeskrivelse**

### **2.1 Innledning**

Den teknologiske utviklingen de siste 10 – 20 årene har ført til en kraftig utvikling av applikasjoner og tjenester som skal hjelpe oss til en lettere og bedre hverdag. Tidligere ble datamaskiner brukt av ingeniører og teknologer for å utføre kalkulasjoner og regnestykker, mens i dag finnes det verktøy som kan assistere de aller fleste samfunnsgrupper i langt flere sammenhenger og med langt flere problemstillinger enn tidligere. Mulighetene for mobilitet øker etterhvert som telenettene blir stadig mer avansert, og flere og flere oppdager at de kan bruke telefonen til mer enn telefonsamtaler. Fremtidens telekommunikasjon vil trolig tilby nye bruksområder for mobile terminaler og applikasjoner for mobile enheter.

Denne oppgaven går ut på å se på mulighetene for å lage en WAP tjeneste som kan brukes som et verktøy for sanntidskoordinering. Jeg vil først ta for meg teorier og forskning innen koordineringsteknologi og bruke dette videre til å avgjøre om det lar seg gjøre å lage en WAP tjeneste for sanntidskoordinering. Til slutt kommer jeg til å lage en prototype/demonstrator dersom det viser seg at teknologien kan brukes til å løse et slikt problem.

## 2.2 Koordinering

Koordinering og koordineringsteori er begreper som man intuitivt har en formening om hva betyr, men det kan være vanskelig å komme med konkrete definisjoner av begrepene. Det er også derfor det verserer svært mange ulike definisjoner på hva koordinering er. En enkel og ukomplisert definisjon er:

”The act of working together” (Malone & Crowston, 1991)

En annen definisjon er:

”Coordination is managing dependancies between activities.” (Malone & Crowston, 1993)

Alle organiserte aktiviteter består av to grunnleggende prosesser (Schieflo & Syvertsen, 1998). Det er deling av aktiviteter mellom deltakerne og koordinering av de resulterende underaktivitetene slik at bestemte mål nåes så effektivt som mulig.

Et av de viktigste kjennetegnene for sterkt koordinerte organisasjoner er at selve koordineringen i seg selv er omtrent usynlig. Først når koordineringen feiler blir temaet tatt opp som et problem. Eksempler på at koordineringen feiler kan være at gjennomføring av oppgaver tar lengre tid enn nødvendig, og ineffektivitet. Det at koordineringen oppfattes som usynlig betyr ikke at den ikke eksisterer eller at den ikke blir tatt på alvor. Det er ofte slik at effektiv koordinering av komplekse aktiviteter vanligvis er et resultat av bevisste strategiske valg, og innføring av effektiv koordinering i organisasjonens mentalitet eller kultur.

Schieflo og Syvertsen identifiserer videre tre hovedkomponenter i organisatorisk koordinering : mål, aktører og aktiviteter. Ved å se på disse komponentene i et koordinasjons-perspektiv kan en si at de er relatert til hverandre gjennom gjensidige avhengigheter. Avhengighetene er logiske konstruksjoner som er avgjørende for hvordan prosessene utvikler seg over tid.

**Mål** (i denne sammenhengen) er noe en kan oppnå som resultat av organisert innsats.

**Aktiviteter** er handlinger som er nødvendig for å nå et gitt mål.

**Aktører** og aktiviteter henger sammen gjennom gjensidige avhengigheter.

Sammenlignet med en tradisjonell måte å betrakte organisasjoner på kan en si at fokus skifter fra å se på de enkelte komponentene i organisasjonen til å se på relasjoner og avhengigheter mellom komponentene.

Koordinering blir da å håndtere avhengigheter for å oppnå samspill. Tilstanden koordinasjon oppnås når avhengighetsrelasjoner er håndtert slik at de virker sammen (Weiseth, 1998).



I følge Mintzberg er organisering å dele en aktivitet opp i arbeidsoppgaver som skal utføres og koordinering av disse slik at aktiviteten blir gjennomført (Weiseth, 1998). De som skal utføre arbeidsoppgavene er avhengige av hverandre og det gjør det nødvendig å sørge for at den enkeltes utførelse henger sammen med det de andre gjør. Det er dette som er koordinering. Gjennomføring av aktiviteter som overgår enkeltpersoners kapasitet må fordeles på flere. Kapasitetsbegrensninger kan være tid, styrke, kunnskap, penger osv. Vi lever i et gjennomorganisert samfunn der vi deler aktiviteter opp i arbeidsoppgaver og koordinerer dem. Som nevnt tidligere er koordinering å håndtere avhengigheter for å oppnå samspill, og avhengighetene kan betraktes som egenskaper ved relasjoner (Weiseth, 1998). En relasjon kan karakteriseres ved hvor ofte det skjer hendelser, hvor lenge den har vart, hvordan den ble etablert og så videre. I tillegg kan den karakteriseres av avhengighet. Avhengigheter kan altså være av forskjellig slag, og de kan ha varierende styrke.

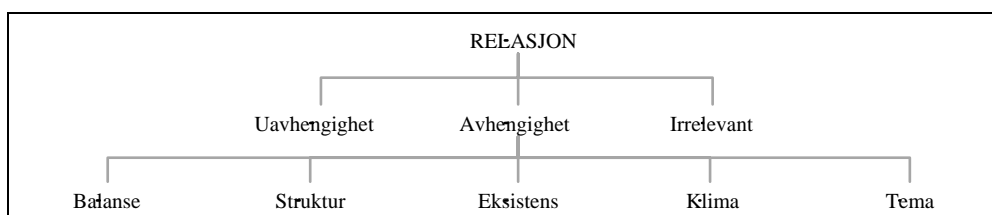
Siden koordinering er noe vi gjør for å oppnå samspill eller samvirke betyr det at vi håndterer en avhengighet for å oppnå samspill. Tilstanden koordinasjon oppnås når avhengighetsrelasjoner er håndtert slik at de virker sammen. Om dette samvirket gjør at aktiviteten gir et godt resultat avhenger av andre forhold i tillegg til graden av koordinasjon. Resultatet av at avhengighetsrelasjoner er håndtert kan altså være en tilstand av koordinasjon, og det er interessant å forstå hva som karakteriserer en slik tilstand. Koordinering er prosessen fram mot tilstanden koordinasjon, på samme måte som organisering er prosessen fram mot tilstanden organisasjon. Avhengighet, koordinering og koordinasjon er sentrale begreper i denne definisjonen fra Weiseth.



Figuren til venstre visualiserer hvordan Weiseth sier at man har en avhengighet og at man håndterer avhengigheten for å oppnå koordinasjon

## 2.2.1 Avhengigheter

Weiseth (1998) har utarbeidet fem egenskapskategorier av avhengighet. De fem kategoriene er avhengighetens balanse, struktur, klima, eksistens og tema. Han skriver videre at en relasjon som er i en tilstand av avhengighet kan beskrives med dens balanse, struktur, eksistens, klima og tema. Dette kan illustreres med følgende figur :

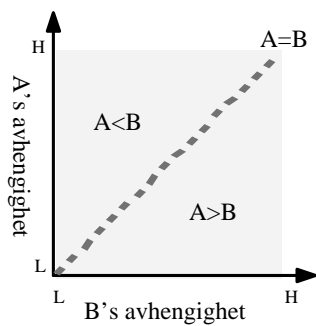


Figur 1 En relasjons tilstand av avhengighet og relasjonens tilstand (Weiseth, 1998)

Videre i rapporten omtales hver av disse fem kategoriene av egenskapene til en avhengighetsrelasjon for så å bruke dette som et rammeverk for den videre drøftingen av problemstillingen rundt ”kompiskjøring”/ kjørelag.

### Avhengighetens balanse

Avhengighet er et generelt uttrykk og uttrykker hvordan aktører er avhengige av hverandre. Vi kan ha gjensidig avhengighet der A er avhengig av B, og B er avhengig av A, men det kan også oppleves at A er avhengig av B, mens B ikke er avhengig av A i det hele tatt. Selv ved gjensidig avhengighet kan det være slik at graden av avhengighet mellom A og B ikke nødvendigvis er lik hos begge parter. Man har da et ubalansert forhold. Er det slik at A og B er like mye avhengig av hverandre kan forholdet karakteriseres som balansert.



De to dimensjonene i figuren representerer henholdsvis A og B sin avhengighet. Deres gjensidige avhengighet er balansert langs den diagonale linjen. På ethvert punkt i området over linjen er B mer avhengig av A, enn motsatt. Tilsvarende, i området under linjen er A mer avhengig av B. I det området som er markert grått, er avhengigheten derfor ubalansert. Det er følgelig grunn til å anta at avhengighet i de fleste tilfeller er ubalansert (Weiseth, 1998).

### Avhengighetens struktur

En avhengighets struktur kan karakteriseres med hvordan en avhengighetsrelasjon kobler sammen to eller flere enheter med tanke på tid og rom.

Weick bruker personer i sine eksempler og sier at han snakker om koblinger generelt (Weiseth, 1998). Weick beskriver *mutual* avhengighet og *sequential* avhengighet {gjensidig, sekvensiell}. Når en person er avhengig av flere personer på samme tid og har indirekte kontroll over sine egne resultater, er det en gjensidig avhengighet. En svakere form for avhengighet er sekvensiell, hvor en person har delvis kontroll over sin egen skjebne og er avhengig av og kan influere på skjebnen til bare én annen person. En annen side ved sekvensiell avhengighet er at en person må formidle gjennom en annen for å få det han ønsker fordi han kan ikke tilby noe til personen han ønsker noe i fra.

### Avhengighetens eksistens

Weiseth sier om avhengigheters eksistens at en avhengighets eksistens beskriver hvem eller hvor en avhengighet er erkjent. Noen personer kan være klar over en avhengighet og andre ikke. En person i en enhet kan være klar over en avhengighet,

selv om hans enhet som sådan ikke er klar over den. Det kan også være motsatt, at enheten har erkjent avhengigheten, men at en av personene i enheten ikke har det. En avhengighet kan erkjennes på et bestemt nivå i organisasjonshierarkiet og ikke på lavere eller høyere nivå. Noe forenklet kan dette beskrives slik at det er et spørsmål om en avhengighet virkelig eksisterer eller ikke. Det kan være slik at våre handlinger utføres ut ifra forutinntatte oppfatninger om hvordan ting skal gjøres, og ikke ut ifra en virkelig relasjon eller avhengighet til andre aktører. Eksempler på slike oppfatninger kan være at man utfører en handling basert på pliktfølelse eller folkeskikk.

### **Avhengighetens klima**

Med avhengighetens klima kan vi forstå hvilke omgivelser en befinner seg i. Man kan tenke seg at to aktører opplever avhengigheten mellom dem som enten fremmede, eller konkurrerende. Fremmede avhengighet er et spill der begge aktørene vinner hvis de samarbeider, mens konkurrerende avhengighet er et spill hvor den ene aktøren bare kan vinne hvis den andre taper. Det er også en forskjell dersom aktørene prøver å nå individuelle eller felles mål med samarbeidet.

### **Avhengighetens tema**

En avhengighetsrelasjon kan bli beskrevet av temaet eller emnet den gjelder. Temaet kan for eksempel være bytte av klienter eller tjenester, deling av ressurser som finansielle ressurser, personale, og materiell eller bytte/deling av informasjon.

### **Oppsummering**

En avhengighets balanse, struktur, eksistens, klima og tema omfatter et vidt spekter av egenskaper ved en avhengighetsrelasjon og konseptualiseringene i hver kategori bidrar til forståelsen av avhengighet. Størst variasjon har Weiseth funnet for avhengighetens struktur. Tendensen i definisjonene tatt i betraktning, er han ikke overrasket over å finne en vektlegging av formelle strukturer. Ved siden av sin viktighet, skyldes antagelig denne vektleggingen den håndgripelige og åpenbare eksistensen til formelle strukturer. Andre, mindre håndgripelige sider ved avhengighet kan det være vanskeligere å få et grep om, men likevel viktige å forstå.

Egenskap	Beskrivelse
Balanse	Nodenes relative avhengighet til hverandre
Struktur	Avhengighetens mønster
Eksistens	Erkjennelsen av en avhengighets eksistens
Klima	Stemningen mellom nodene i avhengigheten
Tema	Hva avhengigheten gjelder

Med de fem kategoriene av egenskaper kan det gis en grundig beskrivelse av en relasjons tilstand av avhengighet; om avhengigheten er gjensidig og hvor balansert den er; hvordan avhengighetsrelasjonen beveger seg i organisasjonslandskapet; hvorvidt og hvordan avhengigheten er erkjent; hva

som preger forholdet mellom de avhengige partene; og hva avhengigheten gjelder. Settet med egenskapskategorier kan danne grunnlaget for et integrert konseptuelt skjema for avhengighetsegenskaper (Weiseth, 1998).

## 2.2.2 Virtuelle grupper

Informasjonsteknologiens fremgang har ført til en dramatisk nedgang i kostnadene forbundet med å prosessere og distribuere informasjon. Denne nedgangen har igjen ført til endringer for hvordan oppgaver løses i ulike sammenhenger. Det blir tatt i bruk informasjonsteknologi i firmaer der dette tidligere var helt utenkelig fordi man ser at teknologien gir fordeler og nye muligheter. Man opplever at det oppstår virtuelle grupper som kun kommuniserer elektronisk, der alt form for samarbeid foregår i det virtuelle rom.

Mye av litteraturen om virtuelle organisasjoner bygger på to antakelser (Kraut et al. 1998). Den første antakelsen er at virtuelle organisasjoner oppstår for å oppnå fordeler ved å kjøpe varer og tjenester fra spesialiserte produsenter. Disse produsentene er i stand til å produsere varer og tjenester billigere og bedre siden de er sterkt spesialiserte på sitt felt. Den andre antakelsen er at moderne data- og telekommunikasjons-nettverk reduserer kostnadene forbundet med koordinering. Det fører til at firmaer kan oppnå konkurransefordeler ved å kjøpe fra en spesialisert produsent uten at transaksjonskostnadene øker, noe man tradisjonelt har vært vant til ved kjøp av varer og tjenester fra eksterne produsenter.

Bruken av Internett fører til at det oppstår grupper der en kan samarbeide og kommunisere med andre over nettet. Dette kalles virtuelle grupper. Når et nettverk av datamaskiner kobler sammen mennesker eller organisasjoner dannes det et sosialt nettverk. Et sosialt nettverk er et nettverk av mennesker, eller entiteter, sammenbundet av sosiale relasjoner, for eksempel vennskap, ønske om samarbeid eller informasjonsutveksling. I disse gruppene oppstår det relasjoner mellom medlemmene, og medlemmene har forskjellige roller i den virtuelle gruppen. Disse rollene kan være helt forskjellige fra deres roller i "den virkelige" verden. Mye av teorien innen området virtuelle grupper tar for seg smale områder og ser på Internett som et isolert sosialt fenomen uten å ta hensyn til hvordan bruken av Internett passer inn i andre sider ved folks liv. Det er imidlertid klart at medlemmer av virtuelle

grupper også har et sosialt liv med sosiale verdier som også spiller inn i det virtuelle samfunnet (Wellman & Gulia 1997).

I den vestlige verden i dag er det slik at bortsett fra nære familiemedlemmer og venner knytter man kun løse forbindelser med resten av ens nettverk (Wellman & Gulia 1997). Man får alle typer tjenester i sitt nettverk, men må gå til forskjellige kontakter for forskjellige typer tjenester. Det er ikke slik at en forbindelse kan tilby alle tjenester. Det betyr at en har forbindelser mot spesialister på forskjellige områder, og at en vedlikeholder disse forbindelsene slik at en har tilgang til mange ressurser.

Internett har ført trenden med spesialiserte nettverk videre. Oppbyggingen av Internett innbyr til å finne ressurser i ulike virtuelle samfunn. En kan søke etter alle slags ressurser på nettet enklere enn i "det virkelige liv", og man kan søke uansett tid og sted. Ønsker man svar på en spesiell forespørsel er det svært sannsynlig at noen kan hjelpe deg, og det finnes spesialiserte grupper på Internett for de fleste problemstillinger.

Et kjennetegn ved virtuelle grupper er at relasjoner mellom medlemmene er løst knyttet. Gruppemedlemmene som utgjør en virtuell gruppe er ofte geografisk avgrenset, har forskjellig bakgrunn, men er et team som kommuniserer elektronisk. Grunnen til at mange firmaer velger å gå over til en mer virtuell organisasjon er at de da opplever å bli mer dynamiske, siden relasjonene er løst knyttet, og at firmaets arbeid blir mer effektivt. Rutinejobber blir automatisert, og kostnader forbundet med koordinering reduseres. Tid er her en viktig kostnadsfaktor.

En virtuell virksomhet kan være et midlertidig nettverk av uavhengige virksomheter – leverandører, kunder, rivaler, tjenesteytere, som ved hjelp av IT raskt innleder et samarbeid. Dette samarbeidet oppstår for å utnytte en oppstått mulighet for eksempel gjennom deling av kompetanse, kostnader og tilgang til hverandres markeder. En har altså en midlertidig samarbeids konstellasjon som vanligvis ikke har noe hovedkontor og heller ikke noe formelt organisasjonskart. Et effektivt IT-nettverk er en nødvendig betingelse for at en virtuell virksomhet skal kunne etableres. Nettverket er et utgangspunkt for rask gjennomføring av eventuelle forhandlinger forut for en avtaleinngåelse. I en forhandlingssituasjon er det ingen tid å miste, og man kan si at den virtuelle virksomheten er etablert i henhold til en elektronisk kontrakt mellom partene.

Noe av poenget med en virtuell virksomhet er raskt å etablere en "uslåelig" allianse bestående av ulike, uavhengige virksomheter (Christensen, Grønland, Methlie 1998). Ved å sette sammen virksomheter med ulik strategisk kjernekompetanse, er det mulig i en gitt situasjon å oppnå en "best-i-alt" situasjon. Når den bestemte oppgaven er løst, kan alliansen oppløses på en enkel måte. Dette er en av grunnene til at virtuelle organisasjoner kan være en interessant og formålstjenlig organisasjonsløsning for fremtiden. Forutsetningen for å lykkes er imidlertid at det eksisterer et vedvarende og gjensidig tillitsforhold mellom partene i alliansen (Christensen, Grønland, Methlie 1998).

### 2.2.3 Nye aktører i markedet

Koordinering av grupper og mennesker har blitt et nytt markedsområde. Dette er sett ut fra programvareprodusenter som leverer software som gjør koordineringsfunksjonene enklere. Produsentene tjener da penger på produktene de selger til sine kunder. Videre har utviklingen av Internett ført til nye "yrkesgrupper" slike som tjenestetilbydere eller informasjons-tilbydere. Dette er yrkesgrupper som formidler informasjon og/eller tjenester mellom aktører i et marked og de kalles ofte for "cybermediaries" eller "infomediaries". Disse aktørene brukes særlig ved elektronisk handel der de skaffer oversikt over f.eks priser i et bestemt marked. En kan også se for seg at en slik "cybermediary" forhandler med leverandører om priser på spesielle tjenester fordi den representerer en interessant kundegruppe. På denne måten kan en "cybermediary" være med på å presse ned prisen på bestemte varer og tjenester, og dette vil motivere brukerne av tjenesten til å støtte opp om sin "cybermediary". (Sarkar, Butler & Steinfield)

Internett har ført til en eksplosjon av antall mulige valg for forbrukerne. Tidligere var butikkenes vareutvalg begrenset av hvor mye hylleplass en har til rådighet, men med den nye teknologien trenger man ikke ha hylleplass i det hele tatt. Med hjelp av Internett kan man få tilgang til alle produkter lagt ut for salg innen en bestemt kategori. Ettersom antall potensielle valgmuligheter øker blir det stadig vanskeligere, og mer tidkrevende, å orientere seg i markedet. For å gjøre tilværelsen lettere for kjøperen dukker det opp "cybermediary"-tjenester som kartlegger tilgjengelige produkter, og som ofte gjør kvalifiserte vurderinger av hvilket produkt man bør velge. Denne vurderingen kan også gjøres ut ifra hvilke ønsker og forutsetninger kunden har. Dersom tjenesten skal gjøre slike vurderinger må kunden legge inn en profil av seg selv. En leverandør er interessert i å skaffe så mye informasjon om kundene sine som mulig, mens kundene gjerne ønsker å være mest mulig anonyme. Denne "interessekonflikten" har oppstått de siste 10-20 årene gjennom markedsføring gjennom reklamesendinger, telefonsalg og "personlige reklamebrev". Dersom en "cybermediary" får tillit hos sine klienter kan den oppnå at klientene legger inn innholdsrike profiler som kan brukes til å få informasjon om produkter som klienten virkelig er interessert i. Hvis en "cybermediary" har mange klienter innenfor en bestemt markedsgruppe, f.eks reisende som ofte reiser på en bestemt strekning, kan dette brukes til å forhandle frem avtaler hos reiseoperatører på den aktuelle strekningen. (Hagel & Singer, 1999)

## 2.3 Krav til et sanntidssystem eller en sanntidstjeneste

En vanlig oppfatning av et sanntidssystem er at det er et raskt og presist system. Dette trenger imidlertid ikke alltid å være en komplett beskrivelse av et slikt system. Systemet må inneholde et nøyaktig bilde av tilstanden til objektene som systemet skal kontrollere. For at et sanntidssystem skal kunne styre sitt kontrollobjekt må det kunne overholde gitte responstider for kontrolloppgavene. Det er en forutsetning at man kan garantere dataenes gyldighet (eng time validity ) for at man skal kunne garantere responstider for sanntidsopp-gaver. Responstiden vil være en maksimumstid, det vil si at den angir en definert **worst-case** maksimumstid for respons. Videre vil responstiden måtte kunne garanteres under alle omstendigheter, også når andre oppgaver med høyere prioritet tillates å avbryte oppgaven. Sett fra kontrollsystemet må det kunne garantere responstid for alle sanntidsopp-gaver. For sanntidssystemer med harde sanntidskrav vil responstidskravet være absolutt, og systemet må anses som ustabil dersom responstids garantien ikke kan garanteres 100 %. "Best-effort" er ikke godt nok for harde sanntids-systemer, men en del sanntidssystemer kan tåle at enkelte oppgaver går noe utover sitt definerte responstids intervall under gitte omstendigheter. Disse systemene omtales gjerne som sanntidssystemer med "myke" sanntidskrav. (Køyen, 1999)

## 2.4 Hvilke typer tjenester er det behov for ?

I en markedsstudie utført av Nokia (Nokia 1999) scorer kategorien banktjenester høyest av ønskede verdiøkende tjenester blant erfarne mobilbrukere. Videre kommer tjenester som fjernstyring, telefonkatalog, ordliste, email, reisebestilling, værmelding, pizzabestilling, aksjeinformasjon og ulike typer nyheter. Trolig vil det etterhvert som bruken av verdiøkende tjenester i telenettet øker, oppstå krav om stadig mer spesialiserte tjenester. Et kjennetegn for alle disse tjenestene er at de utfører konkrete oppgaver for brukeren.

Studien (Nokia 1999) viser videre at mobile tjenester kan deles inn i tre grunnleggende kategorier :

- 1 **Underholdning.** ( for eksempel vitsetjenester)
- 2 **Informasjonstjenester.** (for eksempel nyhetstjenester)
- 3 **Transaksjonstjenester.** (for eksempel banktjenester)

Punkt 1 og 2 kan sees på som rene innholdstjenester, mens transaksjonstjenester er tjenester som inkluderer spesiell funksjonalitet.

**Underholdningstjenestene** fikk dårligst oppslutning i markedsstudien. Dette er tjenester som kan betegnes som generelle, og de ble også omtalt som de mest ubrukelige tjenestene. Dette er typisk tjenester som gir underholdning på fritiden.

**Informasjonstjenestene** fikk middels oppslutning i markedsstudien og ble sett på som saklige og nyttige tjenester. Disse tjenestene er mindre generelle enn underholdningstjenestene, og de søker etter å dekke mer spesielle behov. Eksempler på slike tjenester er nyheter, værmelding for en spesiell by, aksjeinformasjon og ordlister.

**Transaksjonstjenestene** fikk best oppslutning, og da særlig banktjenester. Disse tjenestene kan utføre daglige gjøremål. En banktjeneste kan foreta overføring av penger, en kan kontrollere temperaturen i huset sitt når man er bortreist, man kan bestille drosje eller kanskje få vite hvor den nærmeste italienske restauranten ligger. Transaksjonstjenestene blir sett på som de mest nyttige personlige tjenestene og dette er de mest spesialiserte tjenestene. Tjenestene brukes til å utføre konkrete oppgaver og det er denne type tjenester som står for mest verdiøkning av mobile tjenester (Nokia 1999).



## 2.5 Wireless Application Protocol (WAP)

### 2.5.1 Hva er WAP ?

WAP er en standard (en protokoll) for måten trådløse informasjons- og telefontjenester sendes til mobiltelefoner, personsøkere, PDAer (personlig digital assistent) og andre trådløse enheter. I dag er WAP den protokollen som ligger an til å bli mest fremtredende for slik kommunikasjon (Wapforum 2000).

Flere store teleselskaper jobbet tidligere med egne proprietære standarder for trådløs kommunikasjon. De bestemte seg etterhvert for å samarbeide slik at de kunne utvikle en universell og felles protokoll. En gruppe selskaper, blant de Ericsson, Motorola, Nokia og Phone.com (tidligere Unwired Planet), dannet Wireless Application Protocol Forum i juni 1997 (Wapforum 2000).

I mars 2000 er rundt 150 selskaper registrert som medlemmer i WAP Forum, mens rundt 75 selskaper er støttemedlemmer (Wapforum 2000). Medlemmene, som representerer forskjellige typer bedrifter, jobber sammen for å utvikle WAP protokollen videre. Forumet har følgende mål:

- å gjøre Internett-innhold og avanserte datatjenester tilgjengelig for trådløse telefoner og andre trådløse enheter
- å utvikle en global standard for en trådløs protokoll som fungerer sammen med alle typer teknologier for trådløse nettverk;
- å gjøre det mulig å lage innhold og applikasjoner som kan brukes med systemene til de fleste tilbyderne av trådløse nettverk og de fleste typer trådløse enheter;
- og å benytte seg av og videreutvikle eksisterende standarder og teknologier der dette er mulig og passer inn.

Sluttbrukeren slipper å velge mellom forskjellige konkurrerende systemer, og kan være sikre på at en WAP-telefoner fra ulike leverandører kan kommunisere med alle tjenester, fordi tjenestene benytter WAP-standard (Wapforum 2000).

WAP Forum er opptatt av at markedet for WAP er annerledes enn markedet for Internett, og at en WAP-bruker har andre forventninger og behov enn en Internett-bruker. I utviklingen av protokollen legger derfor WAP Forum vekt på en rekke punkter, som vil forme produktene som lanseres :

- Det skal være enkelt å bruke WAP: En er gjerne i farta når man har behov for informasjon via en trådløs enhet, og det skjer kanskje mye rundt brukeren når han bruker en WAP-enhet.
- Den som bruker en trådløs enhet, må også ha direkte tilgang til datadelen, på samme måte som du kan bruke mobiltelefonen din direkte uten å måtte vente på tilknytning til tjenesten først. Datatjenestene på en WAP-enhet må også være laget slik at de passer til å bli brukt kort tid – du setter deg ikke til for å ”surfe” på en WAP-telefon i timevis, slik du kanskje gjør på en PC.

- Oppgavene som en WAP-bruker antageligvis vil utføre, vil derimot være ganske spesifikke: sjekke om du har mottatt en viktig epost, sjekke nyheter, trafikkmeldinger eller børsnoteringer, bestille flybillett eller hotellrom, sjekke rutetider, bestille drosje, undersøke banksaldoen din eller lignende.

Markedet for trådløse enheter vokser hurtig – Nokia mener det vil være flere enn en milliard trådløse brukere innen slutten av år 2005 (Nokia 1999 B). Siden markedet blir stort og slagkraftig, kan og vil markedet kreve optimaliserte løsninger. Markedsstudier har vist at hvis håndholdte enheter skal slå an blant massene, så må de ikke bli for dyre, under 149 dollar i USA(Nokia 1999 B ). Derfor må WAP-enhetene kunne gi gode informasjons- og kommunikasjonsmuligheter samtidig som produktene ikke kan være for dyre.

## 2.5.2 Teknologien bak WAP

I utviklingen av WAP har medlemmene av WAP Forum fokusert på løsninger som kan fungere best mulig med de begrensningene som finnes i mobile nettverk og mobile enheter.

Realiteten er at trådløse datanettverk som regel tilbyr mindre båndbredde, mindre stabil forbindelse og mindre forutsigelig tilgjengelighet enn andre nettverk. I tillegg kjennetegnes trådløse, håndholdte enheter ofte av at de har svakere prosessorer enn andre datamaskiner, mindre minne, mindre skjerm og at man ofte benytter andre metoder enn et konvensjonelt tastatur for å registrere data. I tillegg bør en trådløs enhet ha et begrenset strømforbruk.

I tillegg til å fokusere på disse egenskapene, er WAP-teknologien også utviklet for å kunne fungere sammen med vel utprøvd Internett-teknologi.

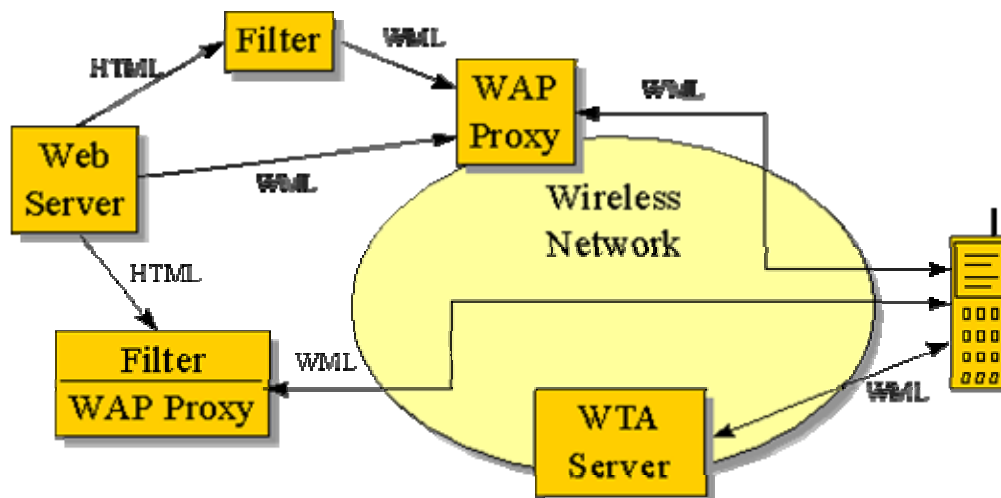
Programmeringsmodellen som blir brukt i WAP er bygget på Internett-løsninger. Målet er at WAP baserte tjenester lett skal kunne utvikles, fordi de som lager nettsider for Internett allerede er kjent med samme type programmer, og dermed relativt lett kan utvikle nettsider for WAP.

WAP sidene blir skrevet i Wireless Markup Language (WML) som er et programmeringsspråk som er utformet for å passe små, mobile enheter. WML ligner på Hypertext Markup Language (HTML).

For å kunne knytte datafunksjoner til WAP-baserte tjenester, benyttes WMLScript. Dette scriptet brukes til å utvikle funksjonaliteten i en tjeneste, omtrent på samme måte som JavaScript brukes i HTML.

WAP-protokollen er bygget på Internett-protokollene som benyttes i dag, Hypertext Transport Protocol (HTTP) og Transport Control Protocol (TCP). Dermed er grunnlaget allerede kjent for utviklerne som jobber med dagens Internett. WML kodes binært når data oversendes til den mobile enheten. En slik koding gjør at datamengden som brukes ved oversending av informasjonen blir så liten som mulig. På denne måten går både overføringen av informasjonen raskere, og det som lastes ned opptar mindre minneplass.

WAP nettverksarkitektur består av komponenter som likner på det vi kjenner fra dagens Internett. WAP terminalen, i dag typisk en mobiltelefon, bruker telefonnettet som medium for kommunikasjon. Overføringshastigheten avhenger derfor av hva dette nettet kan tilby. For at terminalen skal få tilgang til "Internett" kobler den seg til en WAP proxy eller WAP gateway. Herfra går informasjonen videre til en applikasjonsserver med ønsket informasjonsinnhold. En slik applikasjonsserver vil typisk være en vanlig web server som kan svare både på html forespørsler og wml forespørsler.



Figur 2 Oversikt over WAP nettverksarkitektur (source: www.wapforum.org)

### 2.5.3 Wireless Markup Language (WML)

WML er programmeringsspråket som brukes til å beskrive sider som skal vises på WAP enheter. WML tilhører WAP-standarden og brukes på samme måte som HTML brukes for å beskrive tradisjonelle Internett sider. Det er ventet at WAP og WML vil revolusjonere bruksmåter for mobile enheter. Et eksempel på en WAP tjeneste kan være at en er på reise i en by og ønsker å få vite hvor nærmeste McDonalds' restaurant befinner seg. Det letteste vil da være å gå inn på McDonalds' WAP tjeneste (tenkt eksempel) og få den til å fortelle hvor den nærmeste restauranten er. Etterhvert trenger man ikke å oppgi sin posisjon, mobilterminalen og telenettet vet allerede din geografiske plassering og denne informasjonen sendes sammen med forespørselen til en server som returnerer relevante McDonalds adresser. Har man en tilstrekkelig skjerm kan man også få tilsendt bykart om man ønsker det.

Grunnen til at en trenger et alternativ til HTML er den begrensede båndbredde i dagens mobile telenett. I vanlig GSM dataoverføring har man i dag 9.6 kbit/s overføringshastighet og det sier seg selv at da må datamengdene begrenses så mye som mulig. Et alternativ kunne vært å fjerne all grafikk i HTML-sidene, men det er

ikke en god nok løsning. Måten WAP begrenser datamengden på er ved å oversette WML sider, skrevet i vanlig tekstformat, til et binært format. På denne måten kan kapasiteten nesten fordobles. Videre er protokollstakken designet for radiobærer med lav bitrate og stor forsinkelse. Dette gir en bedre ytelse enn TCP/IP i mobile nettverk. Gateway arkitekturen gjør det mulig å konvertere WAP stakken til kjente Internett protokoller på server siden. Dette gjør at applikasjoner og tjenester kan ligge på vanlige web-servere og man kan bruke eksisterende design metoder. I tillegg er det også mulighet for komprimering av datamengden over luftgrensesnittet.

Oppkoblingstiden for en mobil terminal er i dag 15 – 20 sekunder. For å begrense antall oppkoblinger, er WML sidene bygget opp som en slags kortstokk. For hver gang telefonen kobler seg opp mot en tjeneste laster den ned en kortstokk, ”deck of cards”, med et antall kort, ”cards”. Hvert slikt kort representerer en wml-side og på denne måten kan en laste mange sider om gangen som ligger sammen i en korstokk. Det typiske er da at en laster ned et antall sider slik at brukeren kan utføre bestemte handlinger. Input fra brukeren kan så sendes til serveren og på grunnlag av dette kan det genereres en ny korstokk med nye kort. For begrense trafikken ytterligere er det definert et eget script språk som kan utføre handlinger i den mobile terminalen. Dette scriptspråket kalles WML-script.

#### 2.5.4 WML-script

Ren WML-kode har den ulempen at den er forholdsvis statisk. Det betyr blant annet at verdier som mottas via tastetrykk må sendes til en server for å avgjøre om verdiene er korrekte. For å unngå dette er WML-script blitt innført. Oppbyggingen av WML-script minner mye om JavaScript, men det er allikevel vesentlige forskjeller. WML-script tilbyr ikke alle fasiliteter man er vant til fra JavaScript, ganske enkelt fordi det ikke er bruk dem. Til gjengjeld byr språket på en del utvidelser, som hovedsakelig ligger i språkets standardbiblioteker, der det åpnes for mange muligheter man ikke kjenner fra JavaScript. Eksempler på dette kan være at man har mulighet for å lese informasjon fra SIM kortet eller manipulere telefonens adressebok. Dette er unødvendige funksjoner på en stasjonær datamaskin, men meget anvendelige funksjoner når plattformen er en vanlig WAP-telefon.

Språket er bygget opp av en rekke forskjellige elementer som variabler, funksjoner og kontrollstrukturer som angir programmets utførelse. Ut over disse konstruksjoner finnes det et standardbibliotek, med mange underbibliotek, slik at en programmerer ikke trenger å finne opp de mest grunnleggende funksjonene hver gang et nytt program skrives.

Det er særlig funksjoner som tar seg av validering som er viktige for WML-script. Grunnen til dette er at man ved bruk av WML-script kan foreta validering av input lokalt, noe man ellers måtte ha en server for å ta seg av. Når man er koblet opp via en langsom og relativt dyr forbindelse gjelder det å foreta seg så mye som mulig lokalt slik at kommunikasjon med servere begrenses så mye som mulig.

## 2.5.5 Designprinsipper for WAP applikasjoner

Viktige elementer ved design av en WAP applikasjon (Stegavik, Saastad & Grimstad 2000) :

### **Orientering og navigasjon :**

WAP er et begrenset grensesnitt der det er viktig å sørge for at brukerne ikke går seg vill. Dette kan gjøres ved god strukturering av systemet, slik at brukeren hele tiden er der han forventer at han skal være i forhold til oppgaven. I tillegg bør man gi eksplisitt støtte til navigasjon og orientering.

- Forsøk å etablere en trestruktur med et fast toppunkt som det er lett å gå tilbake til.
- Det bør alltid være mulig og lett å gå tilbake ett nivå.
- Det skal alltid være intuitivt hvordan man skal gå til neste side.
- I menystrukturer bør det ikke være mer enn 3 nivåer.
- Antall tastetrykk for å skifte sider og bevege seg mellom nivåer skal minimeres.
- Understreket tekst brukes til å illustrere lenker der brukeren kan gå videre til en annen side. Ved å velge en lenke går man videre til en ny side.
- Når brukeren går fra ett skjermbilde til et annet, bør det gis informasjon om hvor han har kommet.
- Når brukeren får opp et nytt skjermbilde bør det man ser være av en slik karakter at brukeren skjønner at han har kommet til riktig sted. Husk at brukeren i verste fall bare ser de øverste 3-4 linjene.
- Man kan ha mer tekst på et kort enn det som vises på en side (3-4 linjer), men det er begrenset hvor mye det er praktisk å bla igjennom. Hvis informasjonen på et kort overskrider 4-5 skjermmer (16-20 linjer), bør teksten stykkes opp.
- Grupper av informasjon bør skilles med et linjeskift. Dersom dette ikke fungerer kan tre minustegn på en egen linje benyttes. Disse bør sentreres.
- Hvis et inputfelt trenger forklarende tekst bør denne plasseres over feltet.

### **Minimer hukommelsesbelastning**

Alle mennesker har begrenset mental kapasitet. For interaktive applikasjoner er det særlig begrensninger i korttidshukommelse og deling av oppmerksomhet med

omgivelsene som kan være problematisk. Når man sitter ved et skrivebord har man flere muligheter til å omgå disse problemene. Man kan notere litt på en lapp eller gjøre ting saktere. For mobile anvendelser har vi et antall faktorer som gjør ting vanskeligere:

- Komplekst miljø. Vi befinner oss kanskje ute og i bevegelse – dette er en mer komplisert situasjon enn å sitte ved et skrivebord. Forstyrrelser gjør at vi vil ha mindre oppmerksomhet tilgjengelig for å forstå WAP applikasjonen.
- Brukerne vil kanskje oppleve det som om de har større tidspress. Dette stresser brukerne og stjeler oppmerksomhet.
- Det er urealistisk å tenke seg at man skal klare å notere ting mens man bruker applikasjonen.

Vi bør derfor forsøke følgende:

- Det bør ikke være nødvendig å huske koder, tall eller lignende fra en del av applikasjonen til en annen.
- Når brukerne skal velge noe, bør valgene presenteres for dem. De skal ikke behøve å huske hva valgene er.
- I valgsituasjoner bør brukerne til en hver tid presenteres for så enkle problemstillinger som mulig på en gang. Stykk opp problemet hvis overhodet mulig.
- Grensesnittet bør være enklest mulig utformet. Unødvendig informasjon og funksjonalitet skal tas bort.

## **Tilbakemelding**

Når en bruker utfører en handling, bør dette alltid resultere i en tilbakemelding.

## 3 Metode

### 3.1 Innledning

Når det gjelder tjenester for koordinering i sanntid kommer det inn nye krav i forhold til tjenester for koordinering uten et spesielt tidskrav. Det må kartlegges hva slags koordinasjon, og hvilke aktører tjenestesten skal gjelde for.

Sanntidskrav finner en mye innen data og telekommunikasjonsverdenen, men tankegangen kan også brukes i et system for sanntidskoordinering. Også i slike systemer er man avhengig av at systemet gir respons innenfor gitte tidsrammer slik at kontrollobjektene kan styres i ønsket retning.

#### 3.1.1 Krav til en Wap applikasjon

Brukskvalitet er uttrykk for hvilken nytte en bruker har av et produkt eller en tjeneste. Det er viktig å oppnå et godt totalresultat der en tar hensyn til hvordan brukeren oppfatter at han får noe igjen for innsatsen ved bruk av tjenesten.

For informasjonsteknologi generelt er det viktig at de funksjoner som tilbys har høy brukskvalitet. For WAP applikasjoner er dette av flere årsaker spesielt viktig (Stegavik & Saastad 1999) :

Problemløsning. WAP brukere vil ikke bruke mobilterminalene sine til å ”surfe” på Internett. Disse brukerne vil ha små, spesifiserte oppgaver som skal løses raskt. Et eksempel på slike oppgaver er at man ønsker å få vite neste bussavgang eller ønsker å bestille en pizza. Det er derfor viktig å sørge for at brukerne virkelig får løst de oppgavene de har.

Krevende brukssituasjon. I en mobil brukssituasjon vil brukerne trolig oppleve flere forstyrrende faktorer som krever oppmerksomhet. I tillegg har terminalene et svært begrenset brukergrensesnitt å orientere seg i og navigere på. Sammen vil dette kreve at applikasjonen er bedre strukturert og mer gjennomarbeidet, enn applikasjoner vi kjenner fra web, for at den skal fungere.

Brukertester av WAP applikasjoner (Stegavik & Saastad 1999) viser at brukerne er interessert i at det går kjapt og effektivt, og at man får gjort noe nyttig. Designprosessen bør være brukersentrert slik at applikasjoner blir mest mulig nytteorienterte.

### **3.2 Dagens situasjon**

Et kjørelag, i denne sammenhengen, er en gruppe sammensatt av et antall mennesker som deler på å kjøre til og fra arbeid. Motivasjonen for å være med i et slikt kjørelag vil variere fra situasjon til situasjon.

- Miljøbevisst, unngår ”matpakkekjøring”.
- Økonomisk, mindre bilkjøring gir bedre privatøkonomi.
- Finner det hensiktsmessig, rimelig stabil arbeidssituasjon med felles arbeidstid for de som kjører sammen.
- Behagelig å sette seg inn i en bil og halvsove, mens andre tar seg av kjøringen.
- Incentiver eller belønning, dvs at passasjerene betaler en sum for å sitte på.

Hvordan gruppen velger å organisere kjøringen kan avhenge av mange faktorer, men det kan generaliseres tre kategorier ut ifra hvor regelstyrt eller regelbasert samarbeidet er.

#### **3.2.1 Sterk grad av regler**

Et kjørelag basert på sterk grad av regler har ferdig oppsatte regler som medlemmene må forholde seg til. Det kan være at man kjører på et bestemt tidspunkt hver dag, hvert medlem har hver sin faste dag de skal være sjåfør osv. Det kan være slik at dersom man ikke har anledning til å stille opp den dagen man er ansvarlig for, er man allikevel forpliktet til å skaffe transport til de andre i kjørelaget. I slike regelbaserte grupper vil medlemmene til enhver tid vite hva de har å forholde seg til, men det vil være liten grad av dynamikk i gruppen.

#### **3.2.2 Delvis regelbasert**

I et kjørelag delvis basert på regler kan det være slik at en har visse grunnleggende regler eller normer å basere seg på, men samarbeidet fungerer mer dynamisk. Eksempel kan være at medlemmene i kjørelaget har hver sin faste dag de skal kjøre, men dersom den ansvarlige ikke kan stille med bil en dag tar en av de andre seg av kjøringen den aktuelle dagen. Det bør da være en viss form for balanse mellom mengden i kjøring for hvert gruppe-medlem slik at ikke noen får uforholdsmessig stor del av kjøringen.

#### **3.2.3 Liten grad av regler**

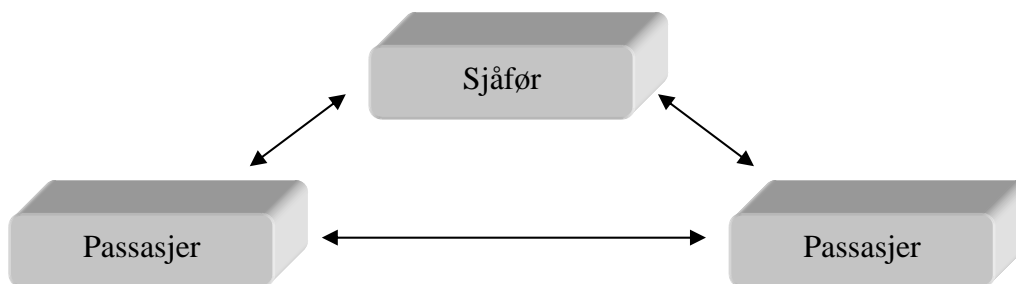
Kjørelag uten særlig grad av faste regler kan være av typen ”-passer det så er jeg med”. I en slik situasjon er det stor grad av frihet for medlemmene. Medlemmene har få forpliktelser, og liten grad av ansvar overfor de andre medlemmene.



### 3.2.4 Aktører

Et kjørelag består av flere deltagere, aktører, som skal utføre en eller flere aktiviteter for å nå sitt mål. Som Schieflo og Syvertsen omtaler henger aktørene og aktivitetene sammen av gjensidige avhengigheter og i følge Weiseth vil tilstanden koordinasjon oppnås når avhengighetsrelasjoner er håndtert slik at de virker sammen. Målet med kjørelaget er at aktørene skal komme seg på jobb.

Den typiske situasjonen ved kjørelag at det er én sjåfør og én eller flere passasjerer. Det vil si at det eksisterer to forskjellige typer aktører, nemlig *sjåføren* og *passasjerer*. For å oppnå koordinasjon må avhengighetsrelasjonene mellom *sjåføren* og *passasjerer* kartlegges, og relasjonene må håndteres til å virke sammen.



Figur 3 Aktører med relasjoner

Figur 3 tar utgangspunkt i at det er tre medlemmer i et kjørelag, to passasjerer og en sjåfør. Pilene mellom aktørene viser hvor det finnes relasjoner. Det er avhengighetsrelasjoner mellom sjåfør og passasjer, og en kan spørre seg om det er avhengighetsrelasjoner mellom hver av passasjerene.

### **3.3 Analyse av avhengighetsrelasjonene**

Som omtalt i kapittel 2.2 har Weiseth utarbeidet fem egenskapskategorier som avhengighetsrelasjoner kartlegges ut ifra, nemlig balanse, struktur, eksistens, klima og tema. For å kartlegge avhengighetsrelasjonene mellom sjåfør-passasjer og passasjer-passasjer kan disse egenskapskategoriene benyttes. Relasjonene er forskjellige avhengig om det er relasjon mellom en sjåfør og en passasjer, eller mellom passasjerer. Relasjonen mellom passasjerene bygger på uavhengighet. Passasjerene er ikke avhengige av hverandre for at målet skal nåes og det er ikke slik at dersom en av passasjerene uteblir vil målet ikke bli nådd, med mindre de andre bestemmer seg for å vente på denne passasjerer.

#### **3.3.1 Avhengighetens balanse**

Avhengighetens balanse er et uttrykk for hvordan aktører er avhengige av hverandre.

##### **Sjåfør og passasjer :**

I relasjonen mellom sjåfør og passasjer er det slik at passasjerer er helt avhengig av sjåføren. Sjåføren er nødt til å være tilstede for at passasjerer skal komme seg til avtalt sted. Sjåføren er imidlertid ikke avhengig av passasjerer i særlig stor grad for å komme seg fram (gitt at sjåføren vet hvor han skal). Det betyr at denne avhengigheten er ubalansert siden den ene aktøren er mer avhengig av den andre.

#### **3.3.2 Avhengighetens struktur**

En avhengighets struktur kan karakteriseres med hvordan en avhengighetsrelasjon kobler sammen to eller flere enheter med tanke på tid og rom.

##### **Sjåfør og passasjer :**

Sjåfør og passasjer har en sekvensiell avhengighet. Passasjerer har delvis kontroll over sin egen skjebne (han må møte opp på avtalt sted til riktig tid), men han er avhengig av sjåføren for å nå sitt mål.

#### **3.3.3 Avhengighetens eksistens**

En avhengighets eksistens går på om en avhengighet er erkjent blant aktørene.

##### **Sjåfør og passasjer :**

Avhengigheten mellom passasjer og sjåfør eksisterer. Passasjerer erkjenner at han er avhengig av sjåføren for å nå sitt mål som er å komme seg til jobb. Sjåføren bør også være klar over at passasjerer er avhengig av ham.

### **3.3.4 Avhengighetens klima**

Med avhengighetens klima kan vi forstå hvilke omgivelser en befinner seg i.

#### **Sjåfør og passasjer :**

Avhengighetens klima mellom sjåfør og passasjer er fremmede. Fremmede avhengighet er et spill der begge aktører vinner hvis de samarbeider. Det vil altså si at både sjåføren og passasjeren vinner på å samarbeide. En av gevinstene ved samarbeid er reduserte transportkostnader for begge aktørene.

### **3.3.5 Avhengighetens tema**

En avhengighet kan bli beskrevet av temaet eller emnet den gjelder. Tema for avhengighetsrelasjonen mellom sjåfør og passasjer er transport. Transporten utføres ved bytte av tjenester, dvs man bytter på å kjøre, og deling av ressurser, dvs flere kjører i samme bil.

## 3.4 Hypotese

### 3.4.1 Innledning

Ut i fra studiene av teknologi beskrevet tidligere i dette kapitlet kan det være mulig å lage en tjeneste i WAP som skal dekke et behov for koordinering. Ut ifra oppgave teksten kan dette være en tjeneste som koordinerer ”kompiskjøring”, et kjørelag, til og fra arbeid i Grimstad.

Det er grunn til å tro at kostnadene forbundet med å organisere og koordinere felleskjøring til og fra arbeid kan reduseres. Funksjoner kan automatiseres slik at man særlig kan gjøre innsparinger i tid. I dagens samfunn er tid en knapphetsfaktor i alle sammenhenger.

### 3.4.2 Scenario

For å sette seg inn i situasjonen til en bruker av denne WAP-tjeneste kan man forestille seg følgende scenario :

I utgangspunktet kan en tenke seg at det er en gruppe som enten trenger skyss, eller som iallfall av og til kjører til jobb, som har tilgang til tjenesten "kompiskjøring". Det kan f.eks. være alle som jobber i Telenor, Ericsson og HiA i Grimstad.

Da vil det være to muligheter for haikeren:

- 1) Er hjemme, står opp og spiser frokost. Ønsker litt fleksibilitet i avreisetidspunktet. Når han er klar til å reise kan han f.eks sende meldingen "NN ønsker avreise fra Vestre Nedenes om 20 minutter". Da ønsker NN å få en tilbakemelding på når han blir plukket opp. Tjenesten må da finne ut om det finnes ledige plasser.
- 2) Er på jobb. En eller annen gang i løpet av dagen ber han tjenesten om skyss hjem for eksempel kl. 15.30, via mail eller ved å gå inn på en Website. Tilsvarende som hjemme, ønsker han nå en bekreftelse på om han har fått skyss, med hvem, hvor han skal møte, tidspunkt etc.

### 3.5 En ny tjeneste blir til

En tjeneste for koordinering av kjøring til og fra arbeid vil i utgangspunktet være svært dynamisk og lite regelbasert [ref kap 3.2.3]. Tanken er at brukeren skal kunne reservere en ledig plass når som helst og det forventes ikke at han må tilby kjøring en annen dag. Som omtalt i kap 3.2 vil motivasjonen for å være med på en slik tjeneste variere, og det er naturlig å spørre seg hvem som stiller opp med ledig plass i bilen.

Avhengighetsrelasjoner mellom aktører er omtalt i kapittel 2.2.1, og analysert i kapittel 3.3, og det fremkommer at det er avhengighetsrelasjoner mellom sjåføren og passasjer(e). Et gjennomgående trekk i analysen av avhengighetsrelasjonene er at det er passasjer(e) som er mest avhengig av sjåføren. Det betyr videre at sjåføren pålegger seg selv et ansvar for å transportere passasjerene ved å tilby plasser i bilen sin. I en sammenheng preget av lite regler vil sjåføren i utgangspunktet ikke få noen form for belønning eller incentiver for det ansvaret han "utsetter" seg selv for. I en lite regelbasert sammenheng kan incentiver være betaling i form av penger som skal dekke sjåførens bensinutgifter. En videreutvikling av tjenesten kan være at telefonabonnementet belastes for en avtalt betalingsavgift, eller at det i telefonens SIM-kort ligger en "lommebok" med elektroniske penger. På denne måten kan passasjerer overføre sine elektroniske penger til sjåførens elektroniske "lommebok". Det vil trolig være vanskelig å holde i gang en tjeneste uten noen form for incentiver. Enkelte vil kanskje være interessert av mer idealistiske grunner, men i det lange løp vil sjåførene trolig ønske en form for betaling eller gjenytelse. Det vil da bli en dreining fra den lite regelbaserte kategorien til den delvis regelbaserte eller den sterkt regelbaserte kategorien. En vil da se at det er visse regler eller normer som ligger til grunn for samarbeidet, for eksempel at sjåføren skal ha en viss sum i avgift for kjøreløp. I mer ekstreme tilfeller kan en tenke seg at WAP tjenesten fører sammen mennesker som ellers ikke ville ha truffet hverandre som danner et eget lukket kjørelag eller en virtuell gruppe (kap 2.2.2). Dette kjørelaget går da bort fra den dynamiske og lite regelbaserte sammenheng, og over til et mer regelstyrt samarbeid slik vi kjenner det fra dagens situasjon med kjørelag. En ser da at WAP tjenesten fungerer mer som et kontaktformidlende verktøy enn et koordinerende verktøy ved at tjenesten brukes for å koble sammen aktører med felles interesser, men ikke så mye koordinere aktørene underveis.

For at kravene til sanntid som omtalt i kap 2.1.4 skal overholdes er det viktig at brukeren innen rimelig tid får beskjed dersom han har fått skyss med noen. I den siste godkjente utgaven av WAP spesifikasjonen, versjon 1.2 godkjent desember 1999, er det spesifisert bruk av push-teknologi som gjør det mulig å sende informasjon fra en WAP server ut til den mobile terminalen. Det er imidlertid ingen produkter som støtter denne funksjonen enda, men et alternativ kan være å benytte seg av SMS-tjenesten, (short messaging services) i situasjoner der det kreves å gi brukerne tilbakemeldinger fra systemet. Eksempler på slike situasjoner vil være dersom en sjåfør av en eller annen grunn er blitt forsinket. Da bør det være mulig for sjåføren å melde fra til tjenesten, slik at hans passasjerer får beskjed om forsinkelsen. Dette er en nødvendig funksjon for at tjenesten skal kunne kalles en sanntids koordineringstjeneste jfr kap 2.3. Det bør også være slik at sjåføren får beskjed hver gang en passasjerer reserverer en plass på turen han har lagt inn i systemet. For at

slike former for interaktivitet skal kunne gjennomføres må systemet/tjenesten til enhver tid kjenne aktørens mobiltelefonnummer. Dette nummeret vil bli brukt til videre kommunikasjon via SMS. Et problem for utvikleren av en tjeneste som dette er at SMS-meldinger er kommunikasjon som teleoperatørene tar seg betalt for å formidle. Det betyr at det vanskelig lar seg gjøre å implementere en tjeneste som baserer seg på kommunikasjon med SMS meldinger uten at en tar seg betalt for dette. Det er imidlertid naturlig at en slik funksjonalitet finnes i et sanntidssystem der kravene til tilbakemelding er tilstede. Push teknologien i neste generasjons WAP vil trolig gi muligheter for større interaktivitet og informasjonsformidling. Med den neste WAP standarden vil det bli mulig å "pushe" informasjon ut til mobilterminalene i motsetning til dagens standard der all informasjon som sendes til mobilterminalen blir sendt på forespørsel fra terminalen.

## **3.6 Applikasjonsløsning**

### **3.6.1 Innledning**

Applikasjonen, eller WAP tjenesten, bør lages med to grensesnitt slik at en kan aksessere tjenesten både fra en WAP terminal og en vanlig nettleser. Det betyr at det må designes både et web grensesnitt og et WAP grensesnitt. Funksjonaliteten bør være lik, men designet kan tilpasses de enkelte grensesnittene.

### **3.6.2 Web grensesnitt**

Webgrensesnittet bør være oppbygget på samme prinsipp som WAP grensesnittet, så enkelt som mulig. Dette er fordi utviklingen av de to grensesnittene skal bli så lik som mulig, og at database kallene kan brukes om igjen. Brukeren må logge seg inn med sitt telefonnummer, og det er dette nummeret som identifiserer hver bruker. Siden oppgaven går ut på å vurdere bruk av WAP som et hjelpemiddel for sanntids koordinering forutsetter jeg at alle som skal bruke tjenesten har tilgang til en mobil enhet som støtter bruk av WAP. Dette medfører at alle brukere av tjenesten har et unikt telefon nummer som gir mulighet for identifisering i systemet. Jeg vil ikke legge så stor vekt på det grafiske i web-brukergrensesnittet da det er selve funksjonaliteten i systemet som skal demonstreres. Brukeren må ha mulighet for å søke etter ledige turer i systemet. Søkekriterier kan være avreisested, ankomststed, dato eller liknende. Webgrensesnittet må også gjøre det så lett som mulig å registrere/legge inn nye turer. Nye turer vil trolig bli lagt inn fra web da man her har tilgang på fullt tastatur og skjerm. Det kan være praktisk å ha mulighet til å legge inn en egen profil i systemet, og hvis en ønsker en slik funksjon er det naturlig at denne ivaretas fra web.

### **3.6.3 WAP grensesnitt**

Som omtalt i kapittel 2.5.5 er brukervennlighet ekstra viktig i en WAP applikasjon på grunn av den begrensede muligheten til å skrive tekst og det lille displayet. WML sider må være så intuitive og selvforklarende at brukeren kan manøvrere seg gjennom applikasjonen uten behov for ekstra hjelpetekst. Med bakgrunn i dette må applikasjonen designes slik at brukeren kan utføre søk etter ledige turer i databasen. Det er naturlig at søkekriteriene er de samme som for web grensesnittet, som avreisested, ankomststed og dato. Det skal også være mulig å registrere/legge inn nye turer fra WAP terminalen. Dette kan bli litt tungvint siden det kreves mange tastetrykk når det er mye informasjon som skal skrives inn fra det begrensede telefontastaturet.

## 4 Utvikling

I dette kapittelet vil jeg ta for meg utviklingen av den konkrete tjenesten.

### 4.1 Utviklingsverktøy

Når en skal utvikle WML-sider kan man strengt tatt nøye seg med en helt enkel tekst editor, for eksempel Notepad. Filen skal kun lages i rent ASCII format. Man vil imidlertid fort støte på problemer fordi man ikke har noen mulighet til å teste koden man har skrevet. For å teste koden kreves det i utgangspunktet at en har en WAP-server og en WAP terminal. Utviklingsverktøyene fra bla Nokia og Ericsson gir imidlertid mulighet for å teste kode ved at de simulerer både en WAP server og en WAP terminal.

Utviklingsverktøy for WAP applikasjoner er fritt tilgjengelig på Internett og disse verktøyene blir stadig forbedret. Jeg har underveis i arbeidet vurdert både Nokia og Ericsson sitt utvikler verktøy men jeg har valgt å benytte meg av Nokia sitt verktøy. Det er dette verktøyet som har fungert best for meg. Nokias programpakke kan lastes ned fra Nokia sine WAP utviklersider på Internett. URL <http://www.nokia.com/wap/index.html>



## 4.2 Programmering

### 4.2.1 Active Server Pages (ASP)

Microsoft Active Server Pages (ASP) er Microsoft sitt rammeverk for å støtte "scripting" på "server-side" nivå. ASP lar deg blande for eksempel HTML eller WML med Jscript eller VBScript og er kompatibelt med Microsoft ActiveX kontroller og komponenter. Selv om du kan bruke ASP til å produsere enkle HTML sider er det særlig utviklet for å knytte sammen web-sider med data lagret i databaser. ASP er ikke et eget script-språk, men det tilbyr et "miljø" på en server som kan prosessere script av ulike typer. ASP brukes som nevnt hovedsakelig til å generere dynamiske web-sider ut fra informasjon som ligger i en database. På denne måten oppnår man at hver gang man oppdaterer sin database blir også web-stedet automatisk oppdatert.

Når det gjelder bruk av WML og script-språk egner ASP seg bra, forutsatt at man har en server som støtter bruk av ASP. Et eksempel er Internet Information Server (IIS) fra Microsoft. Fordelen med ASP er at all kode genereres på serveren før den blir sendt til WAP-terminalen. Man kan da ha et asp-script som på forespørsel aksesserer en database, pakker resultatene inn i wml kode, og sender en gyldig wml-side med et dynamisk innhold tilbake til klienten. Kommunikasjonen mellom asp-scriptet og databasen vil foregå på samme måte uavhengig om det er en forespørsel fra en web-klient eller en WAP-klient. Koden som returneres fra scriptet vil imidlertid avhenge av hvilken klient som har kommet med forespørselen. Det typiske er at man har ett script som returnerer WML-kode og ett script som returnerer HTML-kode.

## 4.3 Selve tjenesten

Under utviklingen av tjenesten har jeg lagt vekt på å følge retningslinjene for design av en WAP applikasjon omtalt i kapittel 2.5.5. Tjenesten lages slik at den kan aksesseres både via et WAP grensesnitt og et web grensesnitt. Fordelen med dette er at en kan bruke webgrensesnittet for mere omfattende operasjoner, f.eks å legge inn nye turer der en må skrive mange tegn, og at en kan bruke WAP terminalen til å søke etter ledige turer og eventuelt reservere turer. Det er også mulig å legge inn nye turer via WAP terminalen, men dette er noe mere tungvint på grunn av begrenset tastatur og display. Siden oppgaven går ut på å lage et eksempel på en tjeneste er det ikke lagt stor vekt på feilhåndtering i programkoden. Grunnen er at slik feilhåndtering ikke vil påvirke oppgavens resultat i særlig grad.

### 4.3.1 Databasestruktur

I ”bunnen” av tjenesten ligger det en Microsoft Access2000 database som inneholder data om alle registrerte turer. Grunnen til at jeg har valgt å bruke Access2000 er at den er tilgjengelig i Microsoft’s Officepakke, og at kartlegging av databaser ikke er en vesentlig del av oppgaven min. Databaseen er bygget opp enkelt av en tabell som inneholder de nødvendige datafelter.

Indeks	ID	Tid	Dato	Arrival	Departure	Plasser	Plass 1-10
4	90123456	12:15	15.06.2000	Arendal	Grimstad	2	Id til hver av passasjerene.

Tabellen gir en oversikt over dataene i databaseen. For å aksessere databaseen utføres SQL-kall fra såkalte ASP-sider ( se kap 4.2.1). Selv om det må utvikles et grensesnitt for WAP og et grensesnitt for web kan de samme SQL kallene til databaseen brukes i begge tilfellene. Det kan være lurt å lage et web grensesnitt først og bruke dette til å sjekke at SQL kallene til databaseen fungerer tilfredsstillende. Web browseren vil ved feil gi en ”litt” konstruktiv tilbakemelding, mens en wml emulator kun sier at det har oppstått en feil. Utvikleren kan derfor spare seg for mange ergrelser ved å først utvikle i et kjent miljø der en kan få tilbakemeldinger på feil som oppstår for så å benytte seg av databasekall han vet fungerer under utvikling av WML applikasjonen.

Dette er en forenklet database uten mulighet for å legge inn egne profiler av brukerne. Grunnen til at dette ikke er implementert er at det medfører mer komplekse SQL kall, men det vil ikke påvirke WAP tjenestens funksjonalitet. En videre utvikling kan være å utvide databaseen med en ny tabell for profiler, og ha en relasjon mellom ID i rute-tabellen og ID i en eventuell profil tabell.

Feltet ID er nøkkelen for å identifisere brukerne i systemet. Dagens WAP standard gir ikke mulighet for å innhente brukerens telefonnummer direkte fra telenettet, slik at brukeren selv må oppgi sitt telefonnummer. Dette er en sikkerhetsrisiko siden det ikke finnes noen mulighet for å verifisere om brukeren oppgir korrekt identitet.

### 4.3.2 Applikasjonsstruktur – Web grensesnitt

Web grensesnittet er bygget opp ved at man først kommer på en side der man registrerer eller ”logger” seg på tjenesten. Siden dette er en WAP tjeneste er det naturlig at brukerne logger seg på med sitt telefonnummer. Dette er et unikt nummer for alle som har en WAP terminal. Videre får man valget mellom å søke etter ledige turer eller å registrere/legge inn nye turer.

#### Søke etter ledige turer : (vedlegg 7 pass2.asp)

På denne siden finnes det tre alternativer.

- 1 Kan søke etter alle ledige biler med ledige plasser.
- 2 Kan søke etter biler fra et bestemt sted.
- 3 Kan søke etter biler fra et bestemt sted på en bestemt dato.

Søkeresultatene kommer i en tabell nederst på siden. I denne tabellen kan en velge en bestemt tur og reservere denne turen.

**Her kan du søke etter turer med ledige plasser.  
Velg alternativ 1, 2 eller 3.**

---

1 Søk etter alle biler med ledige plasser :

---

2 Søk etter biler fra et bestemt sted :

---

3 Reise den :    fra :

---

Velg	Avreisested	Ankomststed	Dato	Tid	Ledige plasser
<input type="radio"/>	Stoa	Fevik	28/4/2000	10:15:00 PM	4
<input type="radio"/>	stoa	grimstad	5/5/2000	12:40:00 PM	2

Figur 4 Søkesiden fra web

### Legge inn tur : (vedlegg 9 calendar.asp)

Ønsker en å legge inn en tur får man opp et skjermbilde der en kan velge turens dato, klokkeslett, avreisested, ankomststed og antall ledige plasser i bilen. For å forenkle denne prosedyren kan datoen velges ved å klikke i en kalender.

**Velkommen**

Her kan du registrere reiseruten din, samt hvor mange ledige plasser du har.

Her kan du velge deg en dato kalenderen skal vise :

Januar 1 2000 Finn dato

May 2000						
Søn	Man	Tir	Ons	Tor	Fre	Lør
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Velg avreisetidspunkt : 07 : 40

Skriv inn avreise- og ankomststed, og antall ledige plasser :

Avreisested : Arendal

Ankomststed : Grimstad

Antall ledige plasser : (1-10) 2

Nullstill Registrer tur

### 4.3.3 Applikasjonsstruktur – WAP/WML grensesnitt

På samme måte som web-grensesnittet starter WAP på en side der man registrerer seg med sitt telefonnummer. En ganske vanlig misforståelse er at man ved bruk av WAP kan identifisere brukeren automatisk med f.eks. telefonnummeret. Telenettet identifiserer WAP terminalen, men denne identifiseringen er ikke tilgjengelig for tjenesteutviklere i dagens WAP standard. Det betyr i vårt tilfelle at brukeren selv må oppgi en form for identifisering overfor systemet.

Et moment en bør passe på er at telefonen ikke står i tekstmodus når det er tall som skal skrives inn. Dette kan styres av applikasjonen. Strukturen og funksjonene er bygget opp på samme måte som beskrevet i foregående kapittel. Fordelen med dette er at databasekallene kan "gjenbrukes" fordi funksjonaliteten mot databasen er lik både for WML og HTML sidene.

Siden WML er bygget opp i et slags "kortstokk-prinsipp" som består en "deck of cards" og et antall "cards" trenger ikke den mobile enheten kontakte serveren annet enn når det kreves prosessering på serveren. For hver gang den mobile enheten

kontakter serveren lastes det ned en "deck" med et antall kort brukeren kan manøvrere seg igjennom før mobilterminalen påny kobler seg til serveren.

Som omtalt i kap 2.5.5 bør det alltid være lett å gå ett nivå tilbake. En måte å løse dette er å lage en "template" med en tilbakeknapp som følger med alle "kortstokkene". På den måten slipper en å lage denne funksjonaliteten på nytt for hvert kort i kortstokken.

Eksempel :

```
<template>
  <do type="prev" name="tilbake" label="Tilbake">
    <prev/>
  </do>
</template>
```

WML delen av tjenesten består av følgende sider :

Deck 1 : start.asp (vedlegg 1 )

Card id: Velkommen

Card id: valg

Card id : sok

Card id: leggin

- finn ledig tur eller legg inn tur

- søk etter tur, kaller Deck 2.

- legg inn ny tur , kaller Deck 3



Eksempel på skjermbilder i WAP tjenesten. Til venstre valg mellom å legge inn tur eller søke etter tur. Til høyre – søkekriterier for å finne ledig tur.

Deck 2: SearchQuery.asp (vedlegg 2 )

Lister ut søkeresultatene som linker og ved en eventuell reservasjon laster deck 4 : wmlres.asp

Deck 3: InputQuery.asp (vedlegg 3 )

Legger tur inn i databasen.

Deck 4 : wmlres.asp (vedlegg 4)

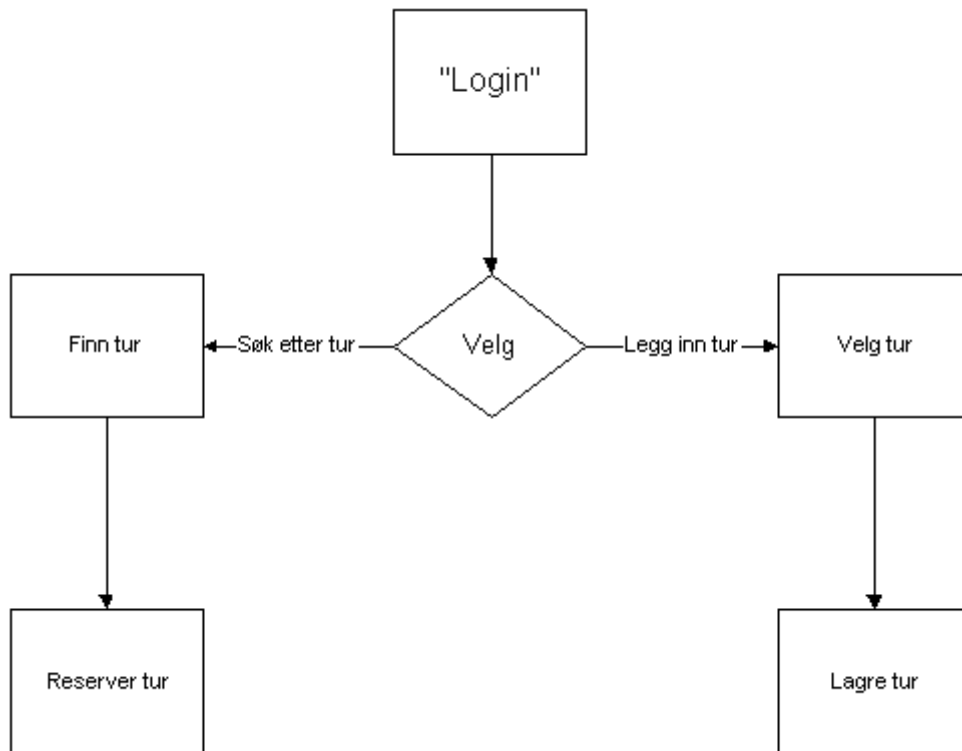
Reserverer en tur i databasen ut i fra brukeren valg i Deck 2.

### Felles filer

Felles fil for både wml-sidene og html-sidene er filen file.dsn (vedlegg 12 ) som er en kobling mellom applikasjonen / tjenesten og databasen. Det er mange måter å opprette en slik kobling på, men ved å bruke en tekstbasert fil som inneholder alle data om hvordan databasen kommuniserer, slipper en å måtte konfigurere serveren på noen måte. Alt en trenger å gjøre er å oppgi i koden hvor denne filen ligger. Applikasjonen vil da benytte seg av databaseinformasjonen fra denne filen for å kommunisere med databasen.

### 4.3.4 Flytskjema

Figuren viser hvordan applikasjonen er bygget opp og hvilke valg som kan gjøres. Applikasjonen er lik både for WAP og web, selv om informasjonen presenteres på forskjellige måter i de to tilfellene.



Figur 5 Flytskjema for WAP applikasjonen

## 5 Resultat

Ut fra det teoretiske studiet av relevante egenskaper og aspekter rundt WAP har jeg funnet at en WAP tjeneste kan benyttes som et verktøy for sanntidskoordinering. Det er imidlertid visse egenskaper ved en slik tjeneste som må være tilstede som det med dagens teknologi ikke er mulig å implementere. Det gjelder særlig for kravene til sanntid. En bruker må få tilbakemelding fra systemet innenfor gitte tidsrammer for at det skal kunne kalles en sanntids tjeneste. Løsninger med tilbakemeldinger kan implementeres med WAP sin "Push"-teknologi som kommer i WAP standard 1.2. Denne standarden ble godkjent i desember 1999 og det forventes at den bli tatt i bruk i løpet av andre halvdel av år 2000 [Kristensen, 2000].

Ut fra begrensningen med mangelfull tilbakemelding har jeg implementert en tjeneste som tar for seg registrering av reiser med ledige plasser samt reservasjon av ledige plasser på reiser. Tjenesten kan aksesseres både fra web og WAP. Funksjonaliteten er lik for begge grensesnittene og den er et utgangspunkt for en tjeneste for sanntidskoordinering. I og med at "Push"-teknologien ennå ikke er implementert i WAP standarden har området rundt tilbakemeldinger mellom aktører og system i sanntid ikke blitt implementert i særlig grad. Brukeren får beskjed når han har registrert turen sin, eller reservert en ledig plass, men utover dette er det ingen kommunikasjon mellom aktørene og systemet. Bruk av SMS gir en mulighet for meldingsutveksling, men dette er tjenester som teleoperatørene tar seg betalt for å tilby, og en kan ikke uten videre sette opp en tjeneste som genererer SMS meldinger.

## 5.1 Wap tjenesten

WAP tjenesten åpner med at brukeren registrerer seg med sitt telefonnummer. Dette nummeret er identifikasjon av brukeren i systemet.



Videre får brukeren valget mellom å legge inn en ny tur i systemet, eller å søke etter ledige turer. Vi antar at brukeren velger å søke etter ledige turer :





Brukeren kan legge inn søkekriterier som avreisested og dato.



Etter utført søk blir søkeresultatene listet opp som linker. Her kan vi se at det er en ledig tur fra Arendal den 18.mai 2000 med 3 plasser ledig klokken 07:25. Dersom brukeren vil være med kan han følge linken, og han vil få reservert en plass på turen. Ønskes et nytt søk velger man "Back" knappen.



Turen er registrert og brukeren har booket seg en plass.

Brukeren kan også legge inn nye turer fra sin WAP terminal :



Brukeren legger da inn aktuelle data for turen og avslutter med å lagre turen i databasen ved å trykke på linken "Lagre tur".

Tjenestens responstid er som ventet ut ifra erfaringer med andre WAP tjenester. Det tar forholdsvis lang tid ( 10 – 20 sekunder ) for terminalen å kontakte serveren og å laste inn sidene, men dette er på grunn av begrenset overføringskapasitet i GSM nettet. Dette vil forbedre seg dramatisk i neste generasjon mobilnett GPRS der man kan være kontinuerlig logget på. Videre tar det lang tid å skrive inn tegn på terminalen på grunn av det begrensede tastaturet. Dette er faktorer som gjør at tidsforbruket ved bruk av tjenesten er ganske stort, særlig hvis man skal registrere nye turer ved hjelp av WAP terminalen. Registreringen av nye turer medfører at en må skrive inn mange tegn i terminalen. Søk etter turer og reservering av turer er enkelt og krever få tastetrykk.

For å forenkle tjenesten mest mulig er søketjenesten laget slik at de mest vanlige avreisestedene er ferdig definert når brukeren skal søke. Det vil si at brukeren slipper å skrive inn for eksempel Arendal, men kan isteden velge stedet ut fra en

forhåndsdefinert liste. Siden denne tjenesten tar for seg kjøring i Grimstad området er dette en grei løsning, men for tjenester som tar for seg større områder bør man heller skrive inn stedsnavnet selv om det er mere tungvint.

En annen forenkling er at man kan forhånds-formatere "input"-felter. Skal man for eksempel skrive en dato, 29/05/2000, kan feltet formateres slik at tegnet "/" kommer opp automatisk og man trenger kun å taste inn 29052000. Denne funksjonaliteten er definert i WAP standarden og fungerer i Nokias utviklingsverktøy, men fungerer ikke i selve telefonen. Det betyr at en utvikler alltid må teste applikasjoner i en ordentlig terminal og ikke kun i en emulator i et utviklingsverktøy.

## 6 Drøfting

Tanken bak et kjørelag som administrerer seg selv ved hjelp av WAP er at organisasjonen blir meget dynamisk og lite regelbasert. Det betyr at hvem som helst kan reservere ledige plasser, og hvem som helst kan tilby plasser i bilen sin. Med denne dynamiske løsningen vil det være vanskelig å få til en rettferdig fordeling av kjøringen siden kjørelagets sammensetning hele tiden varierer. For at en slik dynamisk organisasjon skal opprettholdes vil det sannsynligvis være nødvendig med former for incentiver til sjåførene. På den måten vil det for noen være attraktivt å kjøre ofte dersom det ligger former for betaling til sjåføren i systemet. Tjenesten vil da grense over mot en taxisentral der en kan registrere sine egne turer og ta betalt fra passasjerene som sitter på. Dette vil være en form for ”cybermediary”, omtalt i kap 2.2.3 , som er et virtuelt mellomledd som formidler tjenester mellom, i dette tilfellet, sjåfører og passasjerer.

En annen utvikling en kan tenke seg er at en gruppe blir introdusert for hverandre gjennom WAP tjenesten. Denne gruppen bestemmer seg for å lage sitt eget kjørelag som da blir et utspring fra den større gruppen som omfatter alle deltagerne på WAP tjenesten. Dersom denne utbrytergruppen bestemmer seg for å lage sitt eget kjørelag vil de etter all sannsynlighet danne det på grunnlag av visse regler de bestemmer seg for på ”utsiden” av WAP tjenesten. Et eksempel er at gruppen blir enige om at det skal være en bestemt fordeling på kjøremengden, faste kjøredager etc (jfr kap 3.2). I en slik situasjon vil ikke lenger WAP tjenesten stå i sentrum for koordinering av kjøringen, men heller være et kontaktformidlende hjelpemiddel som hjalp til med etableringen av kjørelaget.

Siden dagens WAP standard ikke støtter bruk av ”push”-teknologi foregår det ikke meldingsutveksling mellom systemet og brukeren bortsett fra tilbakemeldinger brukeren får etter utførte transaksjoner/handlinger. Siden all informasjon kun sendes til brukeren på forespørsel vil aldri brukeren bli oppdatert uten først å etterspørre en slik oppdatering. Det betyr at tjenesten slik den er implementert med dagens teknologi er en tjeneste for koordinering, men ikke i sanntid.

WAP teknologien er fremdeles på ”nybegynner stadiet” og preges til en viss grad av det. At brukeren selv må logge seg på med sitt eget telefon nummer virker tungvint, og det er også en sikkerhetsrisiko da dette er den eneste form for identifisering en har overfor systemet ( kap 4.3.1). Det finnes ingen måte å verifisere terminalens identitet. Senere versjoner av WAP standarden vil gjøre det mulig for tjenesteutviklere å identifisere terminaler direkte. Utvikleren får da tilgang til å bruke terminalens identitet fra telefonnettet og kan på den måten foreta en sikrere identifikasjon av terminalen.

En annen ulempe med dagens teknologi er at terminalene har et lite display og at det tar forholdsvis lang tid hver gang terminalen skal koble seg til en tjenestetilbyder. For å bruke WAP tjenester kreves det mange tastetrykk og mye venting for å oppnå ønsket resultat. Dette er problemer brukerne må leve med inntil neste generasjons mobiltelefon nettverk blir introdusert, General Packet Radio Switched network,

GPRS. I motsetning til dagens linjesvitsjede telenett er GPRS pakkesvitsjet noe som fører til blant annet raskere dataoverføring. Den kanskje mest revolusjonerende nyheten med GPRS er at abonnenten skal betale for den mengden data (antall kilobyte) som transporteres istedenfor den tiden (antall sekunder) han er oppkoblet. Det medfører at terminalen kan en kontinuerlig opprettet forbindelse (Always On), men slipper å betale for annet enn mengden data som overføres. Raskere dataoverføring medfører at det kan lages mer interaktive og prosessorkevende applikasjoner siden overføringen mellom terminalen og tjenestetilbyderen går svært raskt og transaksjoner utføres på sekundet.

I sammenhenger der det kreves en form for koordinering mellom aktører er det slik at WAP i sin nåværende form kan brukes som et hjelpemiddel til økt koordinasjon. WAP kan fremme bruken av felles tjenester og på visse felter gjøre det enklere å nå felles mål. Etter hvert som WAP teknologien utvikler seg vil en også kunne benytte seg av teknologien til å utvikle tjenester for sanntidskoordinering. Tjenester med krav til sanntid krever imidlertid god interaktivitet mellom tjenesten og brukeren og dette lar seg ennå ikke implementere. I neste versjon av WAP standarden blir det mulig å ”pushe”/sende informasjon ut til WAPterminalene. Brukeren vil da innefor visse tidsrammer kunne bli oppdatert så fort ting begynner å endre seg, og brukeren vil med en gang være klar over at en endring har funnet sted. I dag er det slik at brukeren aktivt må forespørre systemet om det har oppstått endringer. Denne funksjonaliteten vil gi bedre støtte for å bruke WAP som et verktøy for koordinering i sann tid.

## 6.1 Et nytt scenario

Transaksjonstjenester som kjennetegnes ved at brukeren gjør spesielle tjenester er etterspurt blant erfarne mobilbrukere (jfr kap 2.4). Det som foreløpig begrenser suksessen til WAP tjenester er begrensninger i dagens WAP standard. I neste generasjon av WAP standarden vil det bli mulighet for ”push”-teknologi som gjør at en terminal kan få tilsendt informasjon ved gitte situasjoner. En slik situasjon kan være at en sjåfør har blitt forsinket, eller at noen har reservert en plass i din bil. Dette vil føre til at koordineringstjenesten kan gå over til å bli en tjeneste for sanntidskoordinering.

Videre vil identifisering av terminaler i telenettet bli tilgjengelig for applikasjonsutviklere som igjen kan bruke dette for sikker identifikasjon av terminaler i tjenesten.

En fremtidsvisjon kan være at systemet hele tiden holder styr på din identitet og hvor du befinner deg. Når du ønsker å reise hjem sier du fra til tjenesten, som da vet hvor du er og hvor du bor, og tjenesten finner en ledig tur for deg. Når sjåføren nærmer seg får du en melding på terminalen din og du kan gjøre deg klar til å gå. Du har nå mulighet til å betale sjåføren direkte med de elektroniske pengene du har lagret i mobilterminalen.

Telefonnettet gir brukeren mulighet til å være kontinuerlig oppkoblet, slik at tidsforbruket reduseres kraftig. Lavere tidsforbruk vil gi større brukervennlighet og flere anvendelsesområder for WAP teknologien.

## 7 Konklusjon

Oppgaven har tatt for seg egenskaper og aspekter rundt WAP for å se på mulighetene for å etablere en tjeneste for sanntidskoordinering. Det er også utviklet en tjeneste som realiserer koordinering.

En definisjon på koordinering er å håndtere gjensidige avhengigheter mellom aktører i et system. Legger en på et tilleggskrav om at tiden også spiller en rolle i systemet kan det defineres et nytt uttrykk kalt sanntidskoordinering.

WAP er en ny teknologi som gjør det mulig å tilby Internett liknende tjenester på små mobile terminaler, for eksempel mobiltelefoner. Disse terminalene kan benyttes overalt der det er dekning for mobiltelefon og det kan lages tjenester som fremmer koordinering og sanntidskoordinering ved hjelp av terminalene. Dagens WAP standard gjør det vanskelig å lage systemer for sanntidskoordinering, men WAP kan være et verktøy for koordinering generelt.

Et kjørelag som koordineres ved hjelp av WAP vil være en dynamisk organisasjon og lite regelbasert. Uten bruk av incentiver til sjåførene vil den dynamiske lite regelbaserte organisasjonen dreie over mot en mer regelstyrt og statisk organisasjon. WAP tjenesten har da fungert som et koordineringsverktøy for å samle aktører med felles mål, men tjenesten er ikke sentral i den videre koordineringen av samarbeidet i kjørelaget.

I neste generasjon av WAP standarden vil det trolig komme funksjonalitet som gjør at teknologien også kan brukes til koordinering i sann tid.

I tiden fremover vil vi se en stor utvikling både innen terminaler og på applikasjonsiden. En viktig grunn til det at interesseorganisasjonen Wapforum jobber for å fremme WAP og har som en av sine målsetninger at WAP skal være enkelt å bruke. Etterhvert som utviklingen går vil en se stadig nye anvendelsesområder innenfor WAP, også innenfor fagområdet sanntidskoordinering.

## 8 Referanser

Christensen, G., Grønland, S.E. & Methlie, L. (1998) *Informasjonsteknologi. Strategi, Organisasjon, Styling*. 2. utgave Bedriftsøkonomens Forlag.

Hagel III, John & Singer, Marc. (1999) *Net Worth, Shaping Markets When Customers Make the Rules*. Harvard Business School Press.

Kraut, R., Steinfield, C., Chan, A., Butler, B. & Hoag, A. (1998 ) *Coordination and Virtualization : The Role of Electronic Networks and Personal Relationships*.  
URL: <http://www.ascusc.org/jcmc/vol3/issue4/kraut.html> [21. februar 2000 ]

Kristensen, Espen. (februar 2000). Ericsson. Personlig meddelelse, møte 18. februar 2000.

Køyen, Geir Myrdahl (1999). *Kort notat om sanntidssystemer og signalering*. Forelesningsnotater utlevert høsten 1999 i faget Signalering i telekommunikasjonssystemer.

Malone, Yates & Benjamin (1987) *Electronic markets and electronic hierarchies*.  
URL:  
<http://riss.keris.or.kr:8080/pubs/citations/journals/cacm/1987-30-6/p484-malone/>  
[21. februar 2000 ]

Malone, T. W. & Crowston, K. G. (1991). *Toward an interdisciplinary theory of coordination* (Technical report #120). Cambridge, MA: Massachusetts Institute of Technology, Center for Coordination Science.

Malone, T. W. & Crowston, K. (1993). *The Interdisciplinary Study of Coordination*  
URL: <http://fag.grm.hia.no/ikt4200/Litteratur/> [15. mars 2000]

Nokia (1999 A) *The Demand for Mobile Value-Added Services. Study of smart messaging, Market Study*.  
URL: [http://www.forum.nokia.com/library/download/study\\_vas.pdf](http://www.forum.nokia.com/library/download/study_vas.pdf) [16. februar 2000]

Nokia (1999 B) *Wireless Application Protocol – The Corporate Perspective. White paper, march 1999*.  
URL: <http://www.forum.nokia.no> [januar 2000]

Nokia (1999 C) *WML Reference ver 1.1*.  
URL: <http://www.forum.nokia.com> [januar 2000 ]

Sarkar, M. B, Butler, B. & Steinfield, C., *Intermediaries and Cybermediaries: A Continuing Role for Mediating Players in the Electronic Marketplace*.  
URL: <http://www.ascusc.org/jcmc/vol1/issue3/sarkar.html> [17.februar 2000]

Schiefloe, P. M. & Syvertsen, T.G. (1998). *Coordination in knowledge-intensive organizations.*

URL: <http://fag.grm.hia.no/ikt4200/Litteratur/> [15. mars 2000]

Stegavik, Harald, & Saastad, Erik.(1999) *Brukergrensesnitt designretningslinjer for WAP applikasjoner.* Versjon 1.0.

URL: <http://utvikler.mobilinfo.com/Telenor.pdf> [16.februar 2000]

Stegavik, Harald, Saastad, Erik, & Grimstad, Tone.(2000) *Brukergrensesnitt designretningslinjer for WAP applikasjoner.* Versjon 1.1.

Email (16/2/2000) fra Harald Stegavik, Telenor.

Wapforum (2000)

URL: <http://www.wapforum.org>

Wapforum (2000)

URL: <http://www.wapforum.org/what/technical.htm>

Weiseth, P. E., *Konseptualisering av avhengigheter;* Kapittel 3 i doktoravhandling ved institutt for sosiologi og statsvitenskap ved NTNU;

URL: <http://fag.grm.hia.no/ikt4200/Litteratur/> [17.mars 2000]

Wellmann, Barry, & Gulia , Milena. (1997) *Net surfers don't ride alone: Virtual communities as communities.*

URL: <http://www.chass.utoronto.ca/~wellman/links/index.html> [28. februar 2000 ]

## 9 Vedlegg

### 9.1 WML/ASP Filer

#### 9.1.1 Vedlegg 1 : start.asp

```
<%
'send the right MIME type
Response.ContentType = "text/vnd.wap.wml"

%><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<!-- Source Generated by WML Deck Decoder -->

<wml>
<template>
  <do type="prev" name="tilbake" label="Tilbake">
    <prev/>
  </do>
</template>

<card id="Velkommen" title="Velkommen" newcontext="true">
  <p>
    til WAP Biltjenesten
  <br/>
    Skriv telefonnummer : <input format="*N" name="tlfnr" type="text" maxlength="8" />
    <anchor title="Registrer">Registrer nummer<go href="#valg">
  </go></anchor>
  <!--<do type="accept" label="Registrer">
    <go href="#valg">
    <setvar name="tlf" value="$tlfnr" />
  </go>
  </do-->
  <p>
</card>

<card id="valg" title="Velg">
  <p>
    <anchor title="Legg inn tur">Legg inn tur<go href="#legginn">
  </go></anchor>
  <br/>
    <anchor title="Søk">Finn ledig tur<go href="#sok">
  </go></anchor>
  <p>
</card>

<card title="Kjørelag Søkeshid" id="sok">
  <p>
  <select name="sted"><optgroup title="Avreisevalg">
    <option value="NULL">Alle Steder</option>
    <option value="Arendal">Arendal</option>
    <option value="Grimstad">Grimstad</option>
    <option value="Fevik">Fevik</option>
    <option value="Stoa">Stoa</option>
```



```
<!--<option value="annet" onpick="#skrive">Annet</option-->
</optgroup></select>
<select name="dato"><optgroup title="Valg av Dato">
<option value="NULL">Alle Dager</option>
<option value="dag">I dag</option>
<option value="morgen">I morgen</option>
<option value="uke">En uke fra i dag</option>
<option value="mnd">En mnd fra idag</option>
</optgroup></select>

<anchor title="Utfør Søk">Utfør Søk <go href="SearchQuery.asp">
<postfield name="sted" value="$(sted)"/>
<postfield name="dato" value="$(dato)"/>
</go></anchor>
</p>
</card>

<card id="legginn" title="Skriv inn">
<p>
Avreisested : <input name="dep" type="text" maxlength="10" />
Ankomststed : <input name="arr" type="text" maxlength="10" />
Dato : <input name="dato" type="text" format="NN\NN\NNNN" />
Klokkeslett : <input name="klokke" type="text" format="NN\;NN" />
Antall plasser: <input name="plasser" type="text" format="*N" maxlength="10" />

<anchor title="Lagre tur">Lagre tur<go href="InputQuery.asp">
<postfield name="dep" value="$(dep)" />
<postfield name="arr" value="$(arr)" />
<!--<postfield name="dag" value="$(dag)" />
<postfield name="mnd" value="$(mnd)" />
<postfield name="ar" value="$(ar)" />-->
<postfield name="dato" value="$(dato)" />
<postfield name="klokke" value="$(klokke)" />
<postfield name="plasser" value="$(plasser)" />
<postfield name="tlf" value="$(tlfnr)" />
</go></anchor>

</p>
</card>

<card id="skrive" title="Skriv">
<p>
stedsnavn : <input name="sted" type="text" maxlength="10" />
</p>
</card>
</wml>
```

## 9.1.2 Vedlegg 2 : SearchQuery.asp

```
<%
Set conn=Server.CreateObject("ADODB.Connection")
Set myRecord1 = Server.CreateObject("ADODB.Recordset")
'Set myRecord2 = Server.CreateObject("ADODB.Recordset")
'conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
conn.open "filedsn=" & Server.MapPath("file.dsn") & ";DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
'send the right MIME type
Response.ContentType = "text/vnd.wap.wml"

%><?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<template>
  <do type="prev" name="tilbake" label="Tilbake">
    <prev/>
  </do>
</template>
<card id="listall" title="Søkeresultat">
<p>
<%
sted    = request("sted")
dato    = request("dato")
id      = request("idf")
'response.write sted & "-" & dato & "<br/>"

If not sted="NULL" and dato="NULL" Then          'alle datoer, bestemt sted
  mySql1 = "SELECT * FROM Rute WHERE Departure=" & sted & "' AND Plasser>0
ORDER BY Dato ASC"
elseif not dato="NULL" and not sted="NULL" Then 'alle steder, bestemt dato
  if dato="dag" then
    sokedato= date 'now '& month(now) & year(now)
'response.write sokedato
    mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "' AND Dato=" & "#" _
    & sokedato & "#" & "' AND Plasser>0"
'response.write mySql1
  elseif dato="morgen" then
    sokedato= date
    nydato= DateAdd("d",1,sokedato)
'response.write sokedato & nydato
    mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "' AND Dato=" & "#" _
    & nydato & "#" & "' AND Plasser>0"
'response.write mySql1
  elseif dato="uke" then
    sokedato= date
    nydato = DateAdd("d",7,sokedato)
'response.write sokedato & nydato
    mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "' AND Dato>" & "#" _
    & sokedato & "#" & "' AND Dato<" & "#" & nydato & "#"
AND Plasser>0 ORDER BY Dato ASC"
```

```

        'response.write mySql1
    elseif dato="mnd" then
        sokedato= date
        nydato = DateAdd("m",1,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "' AND Dato>" & "#" _
        & sokedato & "#" & " AND Dato<" & "#" & nydato & "#"
AND Plasser>0 ORDER BY Dato ASC"
        'response.write mySql1
    end if
    'mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY
Dato ASC"
elseif not dato="NULL" and sted="NULL" Then    'alle steder, bestemt dato
    if dato="dag" then
        sokedato= date 'now '& month(now) & year(now)
        'response.write sokedato
        mySql1 = "SELECT * FROM Rute WHERE Dato=" & "#" _
        & sokedato & "#" & " AND Plasser>0"
        'response.write mySql1
    elseif dato="morgen" then
        sokedato= date
        nydato= DateAdd("d",1,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Dato=" & "#" _
        & nydato & "#" & " AND Plasser>0"
        'response.write mySql1
    elseif dato="uke" then
        sokedato= date
        nydato = Dateadd("d",7,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Dato>" & "#" _
        & sokedato & "#" & " AND Dato<" & "#" & nydato & "#" AND Plasser>0 ORDER
BY Dato ASC"
        'response.write mySql1
    elseif dato="mnd" then
        sokedato= date
        nydato = Dateadd("m",1,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Dato>" & "#" _
        & sokedato & "#" & " AND Dato<" & "#" & nydato & "#" AND Plasser>0 ORDER
BY Dato ASC"
        'response.write mySql1
    end if
    '
    'mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY Dato ASC"

else
    'Lister alle avganger, Allesteder, alle datoer
    mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY Dato ASC"
end if
'mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY Dato ASC"

set myRecord1 = conn.Execute (mySql1)
if myRecord1.eof or myRecord1.bof then
    Response.write("<br />Ingen ledige turer")
else
    'response.write "<select name=" & "tur" & ">"

```

```

Do while not myRecord1.eof
    'response.write("<option value=" & myRecord1.fields("Indeks") &
'" onpick=" & "wmlres.asp?tur=" & myRecord1.fields("Indeks") & ">" _
    '& myRecord1.fields("Departure") & "-" &
myRecord1.fields("Arrival") & myRecord1.fields("tid") & "</option>" & vbcrLf)
    dag = datepart("d",myRecord1.fields("dato"))
    mnd = datepart("m",myRecord1.fields("dato"))
    aar = datepart("yyyy",myRecord1.fields("dato"))
    norskdato= dag & "/" & mnd & "/" & aar
    'response.write norskdato
    response.write("<a href=" & "wmlres.asp?indeks=" &
myRecord1.fields("Indeks") & ">" & myRecord1.fields("Departure") & "-" &
myRecord1.fields("Arrival") _
    & "," & left(myRecord1.fields("tid"),5) & "," & norskdato _
    & "," & myRecord1.fields("plasser") & " ledig.</a><br/>")

    myRecord1.movenext
loop
    '
    response.write "</select>"
end if
myRecord1.close
Set myRecord1 = Nothing
conn.close
Set conn = nothing
%>

</p>
</card>
<card id="listspes">
<p>
liste</p>
</card>

</wml>

<%
Set conn=Server.CreateObject("ADODB.Connection")
Set myRecord1 = Server.CreateObject("ADODB.Recordset")
'Set myRecord2 = Server.CreateObject("ADODB.Recordset")
'conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
conn.open "filedsn=" & Server.MapPath("file.dsn") & ";DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
'send the right MIME type
Response.ContentType = "text/vnd.wap.wml"

%><?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<template>
    <do type="prev" name="tilbake" label="Tilbake">
        <prev/>
    </do>
</template>
<card id="listall" title="Søkeresultat">
<p>
<%
sted    = request("sted")

```

```
dato = request("dato")
id = request("tlf")
'response.write sted & "-" & dato & "<br/>"

If not sted="NULL" and dato="NULL" Then 'alle datoer, bestemt sted
    mySql1 = "SELECT * FROM Rute WHERE Departure=" & sted & "" AND Plasser>0
ORDER BY Dato ASC"
elseif not dato="NULL" and not sted="NULL" Then 'alle steder, bestemt dato
    if dato="dag" then
        sokedato= date 'now '& month(now) & year(now)
        'response.write sokedato
        mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "" AND Dato=" & "#" _
        & sokedato & "#" & "" AND Plasser>0"
        'response.write mySql1
    elseif dato="morgen" then
        sokedato= date
        nydato= DateAdd("d",1,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "" AND Dato=" & "#" _
        & nydato & "#" & "" AND Plasser>0"
        'response.write mySql1
    elseif dato="uke" then
        sokedato= date
        nydato = DateAdd("d",7,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "" AND Dato>" & "#" _
        & sokedato & "#" & "" AND Dato<" & "#" & nydato & "#"
AND Plasser>0 ORDER BY Dato ASC"
        'response.write mySql1
    elseif dato="mnd" then
        sokedato= date
        nydato = DateAdd("m",1,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Departure="
& sted & "" AND Dato>" & "#" _
        & sokedato & "#" & "" AND Dato<" & "#" & nydato & "#"
AND Plasser>0 ORDER BY Dato ASC"
        'response.write mySql1
    end if
    'mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY
Dato ASC"
elseif not dato="NULL" and sted="NULL" Then 'alle steder, bestemt dato
    if dato="dag" then
        sokedato= date 'now '& month(now) & year(now)
        'response.write sokedato
        mySql1 = "SELECT * FROM Rute WHERE Dato=" & "#" _
        & sokedato & "#" & "" AND Plasser>0"
        'response.write mySql1
    elseif dato="morgen" then
        sokedato= date
        nydato= DateAdd("d",1,sokedato)
        'response.write sokedato & nydato
        mySql1 = "SELECT * FROM Rute WHERE Dato=" & "#" _
        & nydato & "#" & "" AND Plasser>0"
        'response.write mySql1
    elseif dato="uke" then
```

```
sokedato= date
nydato = Dateadd("d",7,sokedato)
'response.write sokedato & nydato
mySql1 = "SELECT * FROM Rute WHERE Dato>" & "#" _
& sokedato & "#" & " AND Dato<" & "#" & nydato & "# AND Plasser>0 ORDER
BY Dato ASC"
'response.write mySql1
elseif dato="mnd" then
sokedato= date
nydato = Dateadd("m",1,sokedato)
'response.write sokedato & nydato
mySql1 = "SELECT * FROM Rute WHERE Dato>" & "#" _
& sokedato & "#" & " AND Dato<" & "#" & nydato & "# AND Plasser>0 ORDER
BY Dato ASC"
'response.write mySql1
end if
'
'mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY Dato ASC"

else
'Lister alle avganger, Allesteder, alle datoer
mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY Dato ASC"
end if
'mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY Dato ASC"

set myRecord1 = conn.Execute (mySql1)
if myRecord1.eof or myRecord1.bof then
Response.write("<br />Ingen ledige turer")
else
'response.write "<select name="" & "tur" & "">"

Do while not myRecord1.eof
'response.write("<option value="" & myRecord1.fields("Indeks") &
"" onpick="" & "wmlres.asp?tur=" & myRecord1.fields("Indeks") & "">" _
'& myRecord1.fields("Departure") & "-" &
myRecord1.fields("Arrival") & myRecord1.fields("tid") & "</option>" & vbcrLf)
dag = datepart("d",myRecord1.fields("dato"))
mnd = datepart("m",myRecord1.fields("dato"))
aar = datepart("yyyy",myRecord1.fields("dato"))
norskdato= dag & "/" & mnd & "/" & aar
'response.write norskdato
response.write("<a href="" & "wmlres.asp?indeks=" &
myRecord1.fields("Indeks") & "">" & myRecord1.fields("Departure") & "-" &
myRecord1.fields("Arrival") _
& "," & left(myRecord1.fields("tid"),5) & "," & norskdato _
& "," & myRecord1.fields("plasser") & " ledig.</a><br/>")

myRecord1.movenext

loop
'response.write "</select>"
end if
myRecord1.close
Set myRecord1 = Nothing
conn.close
Set conn = nothing

%>

</p>
</card>
```

```
<card id="listpes">  
<p>  
liste</p>  
</card>  
  
</wml>
```

### 9.1.3 Vedlegg 3 : InputQuery.asp

```
<%
Set conn=Server.CreateObject("ADODB.Connection")
'Set myRecord1 = Server.CreateObject("ADODB.Recordset")
'Set myRecord2 = Server.CreateObject("ADODB.Recordset")
'conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
conn.open "filedsn=" & Server.MapPath("file.dsn") & ";DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
'send the right MIME type
Response.ContentType = "text/vnd.wap.wml"

%><?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>

<template>
  <do type="prev" name="tilbake" label="Tilbake">
    <prev/>
  </do>
</template>

<card id="database">
<p>
<%
departure      = request("dep")
arrival        = request("arr")
'dag           = request("dag")
'mnd          = request("mnd")
'ar           = request("ar")
dato          = request("dato")
resTime       = request("klokke")
plasser       = request("plasser")
id            = request("tid")
'dato = dag & "/" & mnd & "/" & ar
'response.write departure & "," & arrival & "," & dato & "," & klokke _
'& "," & plasser & id
mySql = "INSERT INTO Rute (Id,Tid,Dato,Departure,Arrival,Plasser) VALUES (" & id & "," &
resTime & "," & dato & "," & departure & "," & arrival & "," & plasser & ");"
conn.Execute (mySql)
conn.close
set conn = nothing
%>
Turen er registrert</p>
</card>

</wml>
```



### 9.1.4 Vedlegg 4 : wmlres.asp

```
<%
'on error resume next
Set conn=Server.CreateObject("ADODB.Connection")
Set myRecord1 = Server.CreateObject("ADODB.Recordset")
'Set myRecord2 = Server.CreateObject("ADODB.Recordset")
'conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
conn.open "filedsn=" & Server.MapPath("file.dsn") & ";DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
'send the right MIME type
Response.ContentType = "text/vnd.wap.wml"

%><?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>

<template>
  <do type="prev" name="tilbake" label="Tilbake">
    <prev/>
  </do>
</template>

<card id="Avslutt" title="Utført">
<p>
<%
id          = "$(tfnr)"
'id         =Cint(id)
tfnr       = 123
indeks = request("indeks")
'response.write indeks
mySql1 = "SELECT * FROM Rute WHERE Indeks=" & indeks
set myRecord1 = conn.Execute (mySql1)
teller=1
do while myRecord1.Fields("Plass" & teller & "")
  teller=teller+1
loop
tempSql= "Plass" & teller & "=" & tfnr
'tempSql= "Plass" & teller & "=$(tfnr)"
mySql1 = "UPDATE Rute SET Plasser=" & plasser-1 & ", " & tempSQL & " WHERE Indeks=" &
indeks
'response.write mySql1
conn.Execute (mySql1)
myRecord1.close
set myRecord1 = nothing

conn.close
set conn = nothing
'response.write id
%>
Turen er registrert !
</p>
</card>
</wml>
```

## 9.2 HTML/ASP filer

### 9.2.1 Vedlegg 5 : default.asp

```
<!-- Created: 24.02.2000 13:25:07 -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD><LINK REL=stylesheet HREF="mystyle.css">
  <META NAME="GENERATOR" Content="ASP Express">
  <META HTTP-EQUIV="Content-Type"CONTENT="text/html;CHARSET=iso-8859-1">
  <TITLE>Biltjenesten - Login</TITLE>
</HEAD>
<BODY>
  <BR>
  <BR>
  <BR>
  <CENTER>
  <h3>Velkommen til biltjenesten "Kompiskjøring"</h3>
  <br><br>Logg deg inn med ditt mobiltelefonnummer.<BR>
  <BR>
  <FORM NAME="userinfo" ACTION="/bruker.asp" Method="GET">
  Telefon nummer : <INPUT NAME="tlfnr" TYPE="Text" >
  <BR>
  <BR>
  <INPUT TYPE=RESET VALUE="Nullstill">
  <INPUT TYPE=SUBMIT VALUE="Login">
  </FORM>
  </CENTER>
</BODY>
</HTML>
```

## 9.2.2 Vedlegg 6 : bruker.asp

```
<!-- Created: 24.02.2000 13:49:21 -->
<! DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD><LINK REL=stylesheet HREF="mystyle.css">
<%
if request.querystring("tfnr")="" then response.redirect("./default.asp")
%>
<META NAME="GENERATOR" Content="ASP Express">
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;CHARSET=iso-8859-1">
<TITLE>Bruker.asp</TITLE>
</HEAD>
<BODY>
<BR>
<BR>
<TABLE align="center" width="80%" class="formater">
<tr><td>
For sjåfører som vil legge inn en tur : <A HREF="./calendar2.asp?tfnr=<%
=Request.QueryString("tfnr") %>" onMouseOver="window.status='Har du ledig plass i bilen skal du
trykke her'; return true" onMouseOut="window.status="; return true">trykk her </A>
</td></tr>
<tr><td>
For passasjerer som ønsker å booke en plass :<A HREF="./pass2.asp?tfnr=<%
=Request.QueryString("tfnr") %>" onMouseOver="window.status='Ønsker du a booke en plass skal
du trykke her'; return true" onMouseOut="window.status="; return true">Trykk her
</A>
</td></tr>
</table>
<!---->
</BODY>
</HTML>
```

### 9.2.3 Vedlegg 7 : pass2.asp

```
<!-- Created: 24.02.2000 14:05:56 -->
<! DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<%
'ON ERROR resume next
if request.querystring("tfnr")="" then response.redirect("./default.asp")
'Finne riktig dato :
if len(request.querystring("month"))>0 then
If IsDate(Request.QueryString("date")) Then
    dDate = CDate(Request.QueryString("date"))
Else
    If IsDate(Request.QueryString("month") & "-" & Request.QueryString("day") & "-" &
Request.QueryString("year")) Then
        dDate = CDate(Request.QueryString("month") & "-" & Request.QueryString("day")
& "-" & Request.QueryString("year"))
    Else
        dDate = Date()
        ' The annoyingly bad solution for those of you running IIS3
        If Len(Request.QueryString("month")) <> 0 Or Len(Request.QueryString("day")) <>
0 Or Len(Request.QueryString("year")) <> 0 Or Len(Request.QueryString("date")) <> 0 Then
            Response.Write "The date you picked was not a valid date. The calendar
was set to today's date.<BR><BR>"
        End If
        ' The elegant solution for those of you running IIS4
        'If Request.QueryString.Count <> 0 Then Response.Write "The date you picked was
not a valid date. The calendar was set to today's date.<BR><BR>"
    End If
End If
End if

Set conn=Server.CreateObject("ADODB.Connection")
Set myRecord1 = Server.CreateObject("ADODB.Recordset")
Set myRecord2 = Server.CreateObject("ADODB.Recordset")
'conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
conn.open "filedsn=" & Server.MapPath("file.dsn") & ";DBQ=" &
Server.MapPath("Kompis_ver1.mdb") _
& ";"
'Lister alle avganger
mySql1 = "SELECT * FROM Rute where Plasser>0 ORDER BY Dato ASC"

'Lister avganger fra et spesielt sted
mySql2 = "SELECT * FROM Rute WHERE Departure="" & request.querystring("fra") & "" AND
Plasser>0 ORDER BY Dato ASC"
set myRecord1 = conn.Execute (mySql1)
set myRecord2 = conn.Execute (mySql2)
%>

<HTML>
<HEAD><LINK REL=stylesheet HREF="mystyle.css">
    <META NAME="GENERATOR" CONTENT="ASP EXPRESS">
    <META HTTP-EQUIV="CONTENT-TYPE"CONTENT="TEXT/HTML;CHARSET=ISO-
8859-1">
    <TITLE>Untitled</TITLE>
</HEAD>
<BODY>
<table width="90%" align="center" border="0">
```

```

<tr>
<th colspan=3 align="center">
    Her kan du søke etter turer med ledige plasser. <br>
    Velg alternativ 1, 2 eller 3.
</th>
</tr>
<tr>
<td colspan=3>
    <br><hr>
</td>
</tr>
<tr>
<th align="left" width="5%" valign="top">1</th>
<td valign="top">
    <FORM NAME="userinfo" ACTION="/pass2.asp" Method="GET">
    Søk etter alle biler med ledige plasser :
    </td>
<td align="right"><input type="hidden" name="til" value="true">
    <INPUT NAME="tlfnr" TYPE=HIDDEN VALUE="<%=
request.querystring("tlfnr")%>" >
    <INPUT TYPE=SUBMIT VALUE="List ledige plasser"
onclick="klikk=true"></form>
</td>
</tr>
<tr>
<td colspan=3>
    <hr>
</td>
</tr>
<tr>
<th align="left" width="5%" valign="top">2</th>
<td valign="top">
    <FORM NAME="userinfo" ACTION="/pass2.asp" Method="GET">
    Søk etter biler fra et bestemt sted : </td>
<td align="right"> <INPUT NAME="tlfnr" TYPE=HIDDEN VALUE="<%=
request.querystring("tlfnr")%>" >
    <input type="text" name="fra" size="24">
    <!--</td>
</tr>
<tr>
<td colspan=2 align="right">-->
    <INPUT TYPE=SUBMIT VALUE="List ledige plasser"> </FORM>
</td>
</tr>
<tr>
<td colspan=3><hr>
</td>
</tr>
<!--</table>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=2> -->
    <tr>
    <th align="left" width="5%" valign="top">3</th>
    <td align="left" valign="top">
    Reise den :</td> <!--</tr>
    <TR>--><TD ALIGN="right" colspan="2">
    <FORM ACTION="/pass2.asp" METHOD=GET>
    <SELECT NAME="day">
        <OPTION VALUE=1>1</OPTION>
        <OPTION VALUE=2>2</OPTION>

```

```
<OPTION VALUE=3>3</OPTION>
<OPTION VALUE=4>4</OPTION>
<OPTION VALUE=5>5</OPTION>
<OPTION VALUE=6>6</OPTION>
<OPTION VALUE=7>7</OPTION>
<OPTION VALUE=8>8</OPTION>
<OPTION VALUE=9>9</OPTION>
<OPTION VALUE=10>10</OPTION>
<OPTION VALUE=11>11</OPTION>
<OPTION VALUE=12>12</OPTION>
<OPTION VALUE=13>13</OPTION>
<OPTION VALUE=14>14</OPTION>
<OPTION VALUE=15>15</OPTION>
<OPTION VALUE=16>16</OPTION>
<OPTION VALUE=17>17</OPTION>
<OPTION VALUE=18>18</OPTION>
<OPTION VALUE=19>19</OPTION>
<OPTION VALUE=20>20</OPTION>
<OPTION VALUE=21>21</OPTION>
<OPTION VALUE=22>22</OPTION>
<OPTION VALUE=23>23</OPTION>
<OPTION VALUE=24>24</OPTION>
<OPTION VALUE=25>25</OPTION>
<OPTION VALUE=26>26</OPTION>
<OPTION VALUE=27>27</OPTION>
<OPTION VALUE=28>28</OPTION>
<OPTION VALUE=29>29</OPTION>
<OPTION VALUE=30>30</OPTION>
<OPTION VALUE=31>31</OPTION>
</SELECT>
<SELECT NAME="month">
  <OPTION VALUE=1>Januar</OPTION>
  <OPTION VALUE=2>Februar</OPTION>
  <OPTION VALUE=3>Mars</OPTION>
  <OPTION VALUE=4>April</OPTION>
  <OPTION VALUE=5>Mai</OPTION>
  <OPTION VALUE=6>Juni</OPTION>
  <OPTION VALUE=7>Juli</OPTION>
  <OPTION VALUE=8>August</OPTION>
  <OPTION VALUE=9>September</OPTION>
  <OPTION VALUE=10>Oktober</OPTION>
  <OPTION VALUE=11>November</OPTION>
  <OPTION VALUE=12>Desember</OPTION>
</SELECT>
<SELECT NAME="year">
  <OPTION VALUE=1990>1990</OPTION>
  <OPTION VALUE=1991>1991</OPTION>
  <OPTION VALUE=1992>1992</OPTION>
  <OPTION VALUE=1993>1993</OPTION>
  <OPTION VALUE=1994>1994</OPTION>
  <OPTION VALUE=1995>1995</OPTION>
  <OPTION VALUE=1996>1996</OPTION>
  <OPTION VALUE=1997>1997</OPTION>
  <OPTION VALUE=1998>1998</OPTION>
  <OPTION VALUE=1999>1999</OPTION>
  <OPTION VALUE=2000 SELECTED>2000</OPTION>
  <OPTION VALUE=2001>2001</OPTION>
  <OPTION VALUE=2002>2002</OPTION>
  <OPTION VALUE=2003>2003</OPTION>
```

```

                <OPTION VALUE=2004>2004</OPTION>
                <OPTION VALUE=2005>2005</OPTION>
                <OPTION VALUE=2006>2006</OPTION>
                <OPTION VALUE=2007>2007</OPTION>
                <OPTION VALUE=2008>2008</OPTION>
                <OPTION VALUE=2009>2009</OPTION>
                <OPTION VALUE=2010>2010</OPTION>
            </SELECT>
            <INPUT NAME="tlfnr" TYPE=HIDDEN VALUE="<%=
request.querystring("tlfnr")%>" >
            fra : <input type="text" name="fradato" size="24">
            <INPUT TYPE="submit" VALUE=" Finn denne reisen ">
            </FORM>
        </TD>
    </TR>
    <tr>
        <td colspan=3><hr>
    </td>
</tr>
</TABLE>

<%
if (request.querystring("til")="true") and (len(request.querystring("fra"))=0) then
    if myRecord1.eof or myRecord1.bof then
        response.write "<center><b>Beklager, ingen ledige turer i databasen</b></center>"
    Else
        Response.write "<table width=" & "90%" & " border=" & "1" & " align=" & "center"
& " bgcolor=" & "#FFcc90" & ">"
        Response.write "<form ACTION=" & "./reserver2.asp" & " Method=" & "POST" &
">"
        Response.write "<tr>"
        Response.write
"<th>Velg</th><th>Avreisested</th><th>Ankomststed</th><th>Dato</th><th>Tid</th><th>Ledige
plasser</th>"
        Response.write "</tr>"
        Do while not myRecord1.EOF
            dag = datepart("d",myRecord1.fields("dato"))
            mnd = datepart("m",myRecord1.fields("dato"))
            aar = datepart("yyyy",myRecord1.fields("dato"))
            norskdato= dag & "/" & mnd & "/" & aar
            response.write "<tr>"
            response.write "<td align=" & "center" & "><input type=" & "radio" & "
value=" & myRecord1.Fields("Indeks") _
            & "s" & myRecord1.Fields("Plasser") & " name=" & "resform" &
"></td><td align=" & "center" & ">" _
            & myRecord1.Fields("Departure") & "</td><td align=" & "center" & ">" _
            & myRecord1.Fields("Arrival") & "</td><td>" & norskdato & "</td><td>"
            -
            & myRecord1.Fields("Tid") & "</td><td>" & myRecord1.Fields("Plasser")
& "</td></tr>"
            myRecord1.MoveNext
        Loop
        Response.write "<tr><td colspan=" & "6" & " align=" & "center" _
        & "><INPUT NAME=" & "tlfnr" & " TYPE=HIDDEN VALUE=" &
request.querystring("tlfnr") _
        & " ><INPUT TYPE=SUBMIT VALUE=" & " Reserver " & "</form></td></tr>"
        Response.write "</table>"
    end if
Elseif len(request.querystring("fra"))>0 then

```

```

    if myRecord2.EOF or myRecord2.BOF then
        response.write "<b><center>Beklager, ingen turer fra " & request.querystring("fra")
    & ".</b></center>"
    else
        Response.write "<table width=" & "90%" & " border=" & "1" & " align=" & "center
    bgcolor=" & "#FFcc90" & ">"
        Response.write "<form ACTION=" & "./reserver2.asp" & " Method=" & "POST" &
">"
        Response.write "<tr>"
        Response.write
"<th>Velg</th><th>Avreisested</th><th>Ankomststed</th><th>Dato</th><th>Tid</th><th>Ledige
plasser</th>"
        Response.write "</tr>"
        do until myRecord2.eof
            dag = datepart("d",myRecord2.fields("dato"))
            mnd = datepart("m",myRecord2.fields("dato"))
            aar = datepart("yyyy",myRecord2.fields("dato"))
            norskdato= dag & "/" & mnd & "/" & aar
            response.write "<tr>"
            response.write "<td align=" & "center" & "><input type=" & "radio" & "
value=" & myRecord2.Fields("Indeks") _
            & "s" & myRecord2.Fields("Plasser") & " name=" & "resform" &
"></td><td align=" & "center" & ">" & myRecord2.Fields("Departure") & "</td><td align=" &
"center" & ">" _
            & myRecord2.Fields("Arrival") & "</td><td align=" & "center" & ">" &
norskdato & "</td><td align=" & "center" & ">" _
            & myRecord2.Fields("Tid") & "</td><td align=" & "center" & ">" &
myRecord2.Fields("Plasser") & "</td></tr>"
            myRecord2.MoveNext
        loop
        Response.write "<tr><td colspan=" & "6" & " align=" & "center" _
        & "><INPUT NAME=" & "tlfnr" & " TYPE=HIDDEN VALUE=" &
request.querystring("tlfnr") _
        & "><INPUT TYPE=SUBMIT VALUE=" & " Reserver " & "</form></td></tr>"
        Response.write "</table>"
        end if
    Elseif (len(request.querystring("fradato"))>0) then
        Set myRecord3 = Server.CreateObject("ADODB.Recordset")
        'Lister avganger fra et spesielt sted og spesiell dato
        mySql3 = "SELECT * FROM Rute WHERE Departure=" &
request.querystring("fradato") & " AND Dato=" & "#" _
        & dDate & "#" & " AND Plasser>0"
        set myRecord3 = conn.Execute (mySql3)
        if myRecord3.EOF or myRecord3.BOF then
            response.write "<b><center>Beklager, ingen turer fra " &
request.querystring("fradato") & " denne dagen</b></center>"
        else
            Response.write "<table width=" & "90%" & " border=" & "1" & " align="
& "center bgcolor=" & "#FFcc90" & ">"
            Response.write "<form ACTION=" & "./reserver2.asp" & " Method=" &
"POST" & ">"
            Response.write "<tr>"
            Response.write
"<th>Velg</th><th>Avreisested</th><th>Ankomststed</th><th>Dato</th><th>Tid</th><th>Ledige
plasser</th>"
            Response.write "</tr>"
            do until myRecord3.eof
                dag = datepart("d",myRecord3.fields("dato"))
                mnd = datepart("m",myRecord3.fields("dato"))
                aar = datepart("yyyy",myRecord3.fields("dato"))

```



```
norskdato= dag & "/" & mnd & "/" & aar
response.write "<tr>"
response.write "<td align=" & "center" & "><input type=" &
"radio" & " value=" & myRecord3.Fields("Indeks") _
& "s" & myRecord3.Fields("Plasser") & " name=" & "resform" &
"></td><td align=" & "center" & ">" & myRecord3.Fields("Departure") & "</td><td align=" &
"center" & ">" _
& myRecord3.Fields("Arrival") & "</td><td align=" & "center" &
">" & norskdato & "</td><td align=" & "center" & ">" _
& myRecord3.Fields("Tid") & "</td><td align=" & "center" & ">"
& myRecord3.Fields("Plasser") & "</td></tr>"
myRecord3.MoveNext
loop
Response.write "<tr><td colspan=" & " 6 " & " align=" & "center" _
& "><INPUT NAME=" & "tlfnr" & " TYPE=HIDDEN VALUE=" &
request.querystring("tlfnr") _
& "><INPUT TYPE=SUBMIT VALUE=" & " Reserver " &
"</form></td></tr>"
Response.write "</table>"
end if
myRecord3.close
set myRecord3 = Nothing
Else
response.write "<b><center>Skriv inn avreisested</center></b>"
End if
myRecord1.close
'response.write myRecord1.state
set myRecord1 = Nothing
myRecord2.close
'response.write myRecord2.state
set myRecord2 = Nothing
conn.close
'response.write conn.state
set conn= nothing

%>
</BODY>
</HTML>
```

## 9.2.4 Vedlegg 8 : reserver2.asp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<%
tlfnr          =request.form("tlfnr")
'sjekker om brukeren er logget inn
if tlfnr="" then response.redirect("./default.asp")

strTemp       =request.form("resform")
temp          =InStr(1,strTemp,"s",1)
indeks        =Left(strTemp, temp-1)
plasser       =mid(strTemp,temp+1)

'response.write strTemp & " " & indeks & " " & plasser

Set conn=Server.CreateObject("ADODB.Connection")
Set myRecord1 = Server.CreateObject("ADODB.Recordset")
'conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
conn.open "filedsn=" & Server.MapPath("file.dsn") & ";DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
mySql1 = "SELECT * FROM Rute WHERE Indeks=" & indeks
set myRecord1 = conn.Execute (mySql1)
teller=1
do while myRecord1.Fields("Plass" & teller & "")
    teller=teller+1
loop
tempSql= "Plass" & teller & "=" & tlfnr
mySql2 = "UPDATE Rute SET Plasser=" & plasser-1 & ", " & tempSQL & " WHERE Indeks=" &
indeks
'response.write mySql2
conn.Execute (mySql2)
myRecord1.close
set myRecord1 = nothing
conn.close
set conn = nothing
%>
<html>
<SCRIPT language="JavaScript">
alert ("Turen er registrert og reservert på deg")
</script>

<head><LINK REL=stylesheet HREF="mystyle.css">
<title>Siste</title>
</head>
<body><center>
<a href="./default.asp"><p class="norm">Tilbake til start</p></a></center>
</body>
</html>
```

## 9.2.5 Vedlegg 9 : Calendar2.asp

```
<html>
<head>
<LINK REL=stylesheet HREF="mystyle.css">
</head>
<body>

<%
if request.querystring("tlfnr")="" then response.redirect("./default.asp")

' ***Begin Function Declaration***
Function GetDaysInMonth(iMonth, iYear)
    Select Case iMonth
        Case 1, 3, 5, 7, 8, 10, 12
            GetDaysInMonth = 31
        Case 4, 6, 9, 11
            GetDaysInMonth = 30
        Case 2
            If IsDate("February 29, " & iYear) Then
                GetDaysInMonth = 29
            Else
                GetDaysInMonth = 28
            End If
        End Select
    End Function

Function GetWeekdayMonthStartsOn(dAnyDayInTheMonth)
    Dim dTemp
    dTemp = DateAdd("d", -(Day(dAnyDayInTheMonth) - 1), dAnyDayInTheMonth)
    GetWeekdayMonthStartsOn = WeekDay(dTemp)
End Function

Function SubtractOneMonth(dDate)
    SubtractOneMonth = DateAdd("m", -1, dDate)
End Function

Function AddOneMonth(dDate)
    AddOneMonth = DateAdd("m", 1, dDate)
End Function
' ***End Function Declaration***

Dim dDate ' Date we're displaying calendar for
Dim iDIM ' Days In Month
Dim iDOW ' Day Of Week that month starts on
Dim iCurrent ' Variable we use to hold current day of month as we write table
Dim iPosition ' Variable we use to hold current position in table

' Get selected date. There are two ways to do this.
' First check if we were passed a full date in RQS("date").
' If so use it, if not look for seperate variables, putting them together into a date.
' Lastly check if the date is valid...if not use today

If IsDate(Request.QueryString("date")) Then
```

```
dDate = CDate(Request.QueryString("date"))
Else
  If IsDate(Request.QueryString("month") & "-" & Request.QueryString("day") & "-" &
Request.QueryString("year")) Then
    dDate = CDate(Request.QueryString("month") & "-" & Request.QueryString("day")
& "-" & Request.QueryString("year"))
  Else
    dDate = Date()
    ' The annoyingly bad solution for those of you running IIS3
    If Len(Request.QueryString("month")) <> 0 Or Len(Request.QueryString("day")) <>
0 Or Len(Request.QueryString("year")) <> 0 Or Len(Request.QueryString("date")) <> 0 Then
      Response.Write "The date you picked was not a valid date. The calendar
was set to today's date.<BR><BR>"
    End If
    ' The elegant solution for those of you running IIS4
    If Request.QueryString.Count <> 0 Then Response.Write "The date you picked was
not a valid date. The calendar was set to today's date.<BR><BR>"
    End If
  End If
End If
```

'Now we've got the date. Now get Days in the chosen month and the day of the week it starts on.  
iDIM = GetDaysInMonth(Month(dDate), Year(dDate))  
iDOW = GetWeekdayMonthStartsOn(dDate)

```
myDep=      request.form("Departure")
myArr=      request.form("Arrival")
myPlass=    request.form("antall")
%>
```

<!-- Ny ytre tabell....-->

```
<table border="0" width="90%" align="center">
  <tr><td colspan="2" align="center"><h3>Velkommen</h3></td></tr>
  <tr><td colspan="2" align="center">Her kan du registrere reiseruten din, samt hvor mange
ledige plasser du har.<hr></td></tr>
```

```
<tr>
```

```
<td align="left"><br>
```

```
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=2>
```

```
<tr><td align="char">
```

```
Her kan du velge deg en dato kalenderen skal vise :</td></tr>
```

```
<TR><TD ALIGN="char">
```

```
<FORM ACTION="/calendar2.asp" METHOD=GET>
```

```
<SELECT NAME="month">
```

```
<OPTION VALUE=1>Januar</OPTION>
```

```
<OPTION VALUE=2>Februar</OPTION>
```

```
<OPTION VALUE=3>Mars</OPTION>
```

```
<OPTION VALUE=4>April</OPTION>
```

```
<OPTION VALUE=5>Mai</OPTION>
```

```
<OPTION VALUE=6>Juni</OPTION>
```

```
<OPTION VALUE=7>Juli</OPTION>
```

```
<OPTION VALUE=8>August</OPTION>
```

```
<OPTION VALUE=9>September</OPTION>
```

```
<OPTION VALUE=10>Oktober</OPTION>
```

```
<OPTION VALUE=11>November</OPTION>
```

```
<OPTION VALUE=12>Desember</OPTION>
```

```
</SELECT>
```

```
<SELECT NAME="day">
```

```
<OPTION VALUE=1>1</OPTION>
```

```
<OPTION VALUE=2>2</OPTION>
```

```
<OPTION VALUE=3>3</OPTION>
```

```
<OPTION VALUE=4>4</OPTION>
<OPTION VALUE=5>5</OPTION>
<OPTION VALUE=6>6</OPTION>
<OPTION VALUE=7>7</OPTION>
<OPTION VALUE=8>8</OPTION>
<OPTION VALUE=9>9</OPTION>
<OPTION VALUE=10>10</OPTION>
<OPTION VALUE=11>11</OPTION>
<OPTION VALUE=12>12</OPTION>
<OPTION VALUE=13>13</OPTION>
<OPTION VALUE=14>14</OPTION>
<OPTION VALUE=15>15</OPTION>
<OPTION VALUE=16>16</OPTION>
<OPTION VALUE=17>17</OPTION>
<OPTION VALUE=18>18</OPTION>
<OPTION VALUE=19>19</OPTION>
<OPTION VALUE=20>20</OPTION>
<OPTION VALUE=21>21</OPTION>
<OPTION VALUE=22>22</OPTION>
<OPTION VALUE=23>23</OPTION>
<OPTION VALUE=24>24</OPTION>
<OPTION VALUE=25>25</OPTION>
<OPTION VALUE=26>26</OPTION>
<OPTION VALUE=27>27</OPTION>
<OPTION VALUE=28>28</OPTION>
<OPTION VALUE=29>29</OPTION>
<OPTION VALUE=30>30</OPTION>
<OPTION VALUE=31>31</OPTION>
</SELECT>
<SELECT NAME="year">
  <OPTION VALUE=1990>1990</OPTION>
  <OPTION VALUE=1991>1991</OPTION>
  <OPTION VALUE=1992>1992</OPTION>
  <OPTION VALUE=1993>1993</OPTION>
  <OPTION VALUE=1994>1994</OPTION>
  <OPTION VALUE=1995>1995</OPTION>
  <OPTION VALUE=1996>1996</OPTION>
  <OPTION VALUE=1997>1997</OPTION>
  <OPTION VALUE=1998>1998</OPTION>
  <OPTION VALUE=1999>1999</OPTION>
  <OPTION VALUE=2000 SELECTED>2000</OPTION>
  <OPTION VALUE=2001>2001</OPTION>
  <OPTION VALUE=2002>2002</OPTION>
  <OPTION VALUE=2003>2003</OPTION>
  <OPTION VALUE=2004>2004</OPTION>
  <OPTION VALUE=2005>2005</OPTION>
  <OPTION VALUE=2006>2006</OPTION>
  <OPTION VALUE=2007>2007</OPTION>
  <OPTION VALUE=2008>2008</OPTION>
  <OPTION VALUE=2009>2009</OPTION>
  <OPTION VALUE=2010>2010</OPTION>
</SELECT>
<INPUT NAME="tlfnr" TYPE=HIDDEN VALUE="<%=
request.querystring("tlfnr")%>" >
<INPUT TYPE="submit" VALUE=" Finn dato ">
</FORM>
</TD>
</TR>
</TABLE>
```

```

<TABLE BORDER=10 CELLSPACING=0 CELLPADDING=0>
<TR><TD>

<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=1
BGCOLOR=#99CCFF>
<TR>
<TD BGCOLOR=#000099 ALIGN="center" COLSPAN=7>
<TABLE WIDTH=100% BORDER=0
CELLSPACING=0 CELLPADDING=0>
<TR>
<TD ALIGN="right"><A
HREF="/calendar2.asp?date=<%= SubtractOneMonth(dDate)
%>&tfnr=<%=request.querystring("tfnr") %>"><FONT COLOR=#FFFF00 SIZE="-
1">&lt;&lt;</FONT></A></TD>
<TD ALIGN="center"><FONT
COLOR=#FFFF00><B><%= MonthName(Month(dDate)) & " " & Year(dDate)
%></B></FONT></TD>
<TD ALIGN="left"><A
HREF="/calendar2.asp?date=<%= AddOneMonth(dDate) %>&tfnr=<%=request.querystring("tfnr")
%>"><FONT COLOR=#FFFF00 SIZE="-1">&gt;&gt;</FONT></A></TD>
</TR>
</TABLE>
</TD>
</TR>
<TR>
<TD ALIGN="center" BGCOLOR=#0000CC><FONT
COLOR=#FFFF00><B>Søn</B></FONT><BR><IMG SRC="/images/spacer.gif" WIDTH=60
HEIGHT=1 BORDER=0></TD>
<TD ALIGN="center" BGCOLOR=#0000CC><FONT
COLOR=#FFFF00><B>Man</B></FONT><BR><IMG SRC="/images/spacer.gif" WIDTH=60
HEIGHT=1 BORDER=0></TD>
<TD ALIGN="center" BGCOLOR=#0000CC><FONT
COLOR=#FFFF00><B>Tir</B></FONT><BR><IMG SRC="/images/spacer.gif" WIDTH=60
HEIGHT=1 BORDER=0></TD>
<TD ALIGN="center" BGCOLOR=#0000CC><FONT
COLOR=#FFFF00><B>Ons</B></FONT><BR><IMG SRC="/images/spacer.gif" WIDTH=60
HEIGHT=1 BORDER=0></TD>
<TD ALIGN="center" BGCOLOR=#0000CC><FONT
COLOR=#FFFF00><B>Tor</B></FONT><BR><IMG SRC="/images/spacer.gif" WIDTH=60
HEIGHT=1 BORDER=0></TD>
<TD ALIGN="center" BGCOLOR=#0000CC><FONT
COLOR=#FFFF00><B>Fre</B></FONT><BR><IMG SRC="/images/spacer.gif" WIDTH=60
HEIGHT=1 BORDER=0></TD>
<TD ALIGN="center" BGCOLOR=#0000CC><FONT
COLOR=#FFFF00><B>Lør</B></FONT><BR><IMG SRC="/images/spacer.gif" WIDTH=60
HEIGHT=1 BORDER=0></TD>
</TR>
<%
' Write spacer cells at beginning of first row if month doesn't start on a Sunday.
If iDOW <> 1 Then
Response.Write vbTab & "<TR>" & vbCrLf
iPosition = 1
Do While iPosition < iDOW
Response.Write vbTab & vbTab & "<TD>&nbsp;&nbsp;&nbsp;</TD>" &
vbCrLf
iPosition = iPosition + 1
Loop
End If

```

```
' Write days of month in proper day slots
iCurrent = 1
iPosition = iDOW
Do While iCurrent <= iDIM
  ' If we're at the beginning of a row then write TR
  If iPosition = 1 Then
    Response.Write vbTab & "<TR>" & vbCrLf
  End If

  ' If the day we're writing is the selected day then highlight it somehow.
  If iCurrent = Day(dDate) Then
    Response.Write vbTab & vbTab & "<TD
BGCOLOR=#FF9730><FONT SIZE=""-1""><B>" & iCurrent & "</B></FONT><BR><BR></TD>"
    & vbCrLf
  Else
    Response.Write vbTab & vbTab & "<TD><A
HREF=""./calendar2.asp?date=" & Month(dDate) & "-" & iCurrent & "-" & Year(dDate) & "&tfnr="
& Request.querystring("tfnr") & """"><FONT SIZE=""-1"">" & iCurrent &
"</FONT></A><BR><BR></TD>" & vbCrLf
  End If

  ' If we're at the end of a row then write /TR
  If iPosition = 7 Then
    Response.Write vbTab & "</TR>" & vbCrLf
    iPosition = 0
  End If

  ' Increment variables
  iCurrent = iCurrent + 1
  iPosition = iPosition + 1
Loop

' Write spacer cells at end of last row if month doesn't end on a Saturday.
If iPosition <> 1 Then
  Do While iPosition <= 7
    Response.Write vbTab & vbTab & "<TD>&nbsp;</TD>" &
vbCrLf

    iPosition = iPosition + 1
  Loop
  Response.Write vbTab & "</TR>" & vbCrLf
End If
%>
</TABLE>
</TD>
</TR>
</table>

<BR>

<td>

<table border="0" cellpadding="2" align="right">
<tr>
<th align="left" valign="top">Velg avreisetidspunkt :</th>
<td align="right" valign="bottom">
<FORM ACTION=""./reserver.asp" METHOD=POST>
<SELECT NAME="iTime">
<OPTION VALUE=0>00</OPTION>
<OPTION VALUE=1>01</OPTION>
```

```

<OPTION VALUE=2>02</OPTION>
<OPTION VALUE=3>03</OPTION>
<OPTION VALUE=4>04</OPTION>
<OPTION VALUE=5>05</OPTION>
<OPTION VALUE=6>06</OPTION>
<OPTION VALUE=7>07</OPTION>
<OPTION VALUE=8>08</OPTION>
<OPTION VALUE=9>09</OPTION>
<OPTION VALUE=10>10</OPTION>
<OPTION VALUE=11>11</OPTION>
<OPTION VALUE=12 SELECTED>12</OPTION>
<OPTION VALUE=13>13</OPTION>
<OPTION VALUE=14>14</OPTION>
<OPTION VALUE=15>15</OPTION>
<OPTION VALUE=16>16</OPTION>
<OPTION VALUE=17>17</OPTION>
<OPTION VALUE=18>18</OPTION>
<OPTION VALUE=19>19</OPTION>
<OPTION VALUE=20>20</OPTION>
<OPTION VALUE=21>21</OPTION>
<OPTION VALUE=22>22</OPTION>
<OPTION VALUE=23>23</OPTION>
<OPTION VALUE=24>24</OPTION>
</SELECT><b> : </b>
<SELECT NAME="iMinutt" >
<OPTION VALUE=00 SELECTED>00</OPTION>
<OPTION VALUE=05>05</OPTION>
<OPTION VALUE=10>10</OPTION>
<OPTION VALUE=15>15</OPTION>
<OPTION VALUE=20>20</OPTION>
<OPTION VALUE=25>25</OPTION>
<OPTION VALUE=30>30</OPTION>
<OPTION VALUE=35>35</OPTION>
<OPTION VALUE=40>40</OPTION>
<OPTION VALUE=45>45</OPTION>
<OPTION VALUE=50>50</OPTION>
<OPTION VALUE=55>55</OPTION>
</SELECT>
<INPUT NAME="tid" TYPE=HIDDEN VALUE="<%= Response.Write(dDate)%>"
<BR><br><br>

</td></tr>
<TR>
<td colspan="2">Skriv inn avreise- og ankomststed, og antall ledige plasser : </td>
</tr>
<tr>
<td><INPUT NAME="tlfnr" TYPE=HIDDEN
VALUE="<%=request.querystring("tlfnr")%">" ></td>
</tr>
<tr>
<th align="left">Avreisested :</th><td align="right"><INPUT NAME="departure"
TYPE="Text" >
</td>
</tr>
<tr>
<th align="left">Ankomststed : </th><td align="right"><INPUT
NAME="arrival" TYPE="Text" value="<%=Response.write(myArr)%>">
</td>
</tr>
<tr>
<th align="left">Antall ledige plasser : (1-10)</th><td align="right"><INPUT
NAME="antall" TYPE="Text" value="<%=Response.write(myPlass)%>">

```



```
</td>
</tr>
<tr><td align="left"> <INPUT TYPE=RESET VALUE="Nullstill"></td><td
align="right"><INPUT TYPE=SUBMIT VALUE=" Registrer tur  " >
</td>
</tr>
</table>
</FORM>
</td>
</TABLE>
</td>
</tr>
</td></tr>
</table>
</TABLE>
</body></html>
```

## 9.2.6 Vedlegg 10 : reserver.asp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Untitled</title>
  <LINK REL=stylesheet HREF="mystyle.css">
</head>
<body>
<p class="norm">
<%
resTime=      request.form("iTime") & ":" & request.form("iMinutt")
dato=         request.form("tid")
plasser=      request.form("antall")
arrival=      request.form("arrival")
departure=    request.form("departure")
ID=           request.form("tlfnr")

if ID="" then response.redirect("./default.asp")

'response.write dato & " "
'response.write resTime & plasser & arrival & departure & ID

Set conn=Server.CreateObject("ADODB.Connection")
'conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
conn.open "filedsn=" & Server.MapPath("file.dsn") & ";DBQ=" &
Server.MapPath("Kompis_ver1.mdb") & ";"
// her må dataene legges inn i databasen

mySql = "INSERT INTO Rute (Id,Tid,Dato,Departure,Arrival,Plasser) VALUES (" & id & "," &
resTime & "," & dato & "," & departure & "," & arrival & "," & plasser & ");"
//mySql = "SELECT * FROM Rute"

'response.write mySql
conn.Execute (mySql)
//set myRs1 = conn.Execute (mySql)
//if myRs1.EOF then
//      response.write "tomt" //myRs1.count
//else
//      response.write myRs1.fields.item(0)
//end if
'set myRs1 = Nothing
conn.close
set conn = nothing
Response.write "<br>Du har lagt inn følgende tur : <br> Fra " & departure & " til " & arrival & " kl "
& resTime & " den " & dato & " med " & plasser & " ledige plasser <br>"
%>
<p class="norm"><br><br>
<a href="./default.asp">Tilbake til start</a>
</p>
</body>
</html>
```

## 9.2.7 Vedlegg 11 : mystyle.css

```
BODY      {
    background-color: #528294;
    /*background-color: #000000;   */
    margin-left: 10em
    margin-right: 4em
    font-family: Comic Sans MS, Sans Serif;
    font-weight: normal;
    font-size: 14;
    text-indent: 5em;
    }

p.avs     {
    font-weight: bold;
    }

p.norm    {
    /*background-color: #FFFFFF;*/
    font-family: Comic Sans MS, Sans Serif;
    font-weight: normal;
    font-size: 14;
    text-indent: 5em;
    }

table.formater {
    font-family: Comic Sans MS, Sans Serif;
    }

td.heading {
    text-align: center;
    /*background-color: #528294;*/
    background-color: #FFFFFF;
    font-size: 20;
    }

td.marg   {
    width: 10%;
    text-align: left;

    /*background-color: #528294;*/
    font-size: 14;
    }

td.vanlig {
    font-size: 14;
    text-indent: 0.5em;
    /*background-color: #FFFFFF;*/
    }
```

## 9.3 Felles filer

### 9.3.1 Vedlegg 12 : file.dsn

```
[ODBC]
DRIVER=Microsoft Access Driver (*.mdb)
ReadOnly=0
UserCommitSync=Yes
Threads=20
SafeTransactions=0
PageTimeout=5
MaxScanRows=8
MaxBufferSize=512
ImplicitCommitSync=Yes
FIL=MS Access
DriverId=25
```