

HOVEDPROSJEKT: **HØGSKOLEN I GJØVIK, VÅREN 2001.**

TITTEL

**PLAYSTATION - PORTAL**

FORFATTERE: \* **KRISTOFFER EMIL AMUNDSEN**  
\* **THORGRIM KRISTIANSEN**

DATO: **23. MAI 2001.**



## SAMMENDRAG AV HOVEDPROSJEKT

<b>Tittel:</b>	<u>PLAYSTATION - PORTAL</u>	<b>Nr. :</b>	
		<b>Dato:</b>	22.05.01
<b>Deltaker(e):</b>	<u>Kristoffer Emil Amundsen, 99dmua</u> <u>Thorgrim Kristiansen, 99dmua</u>		
<b>Veileder(e):</b>	<u>Øivind Kolloen</u>		
<b>Oppdragsgiver:</b>	<u>Arnt Grøstad</u>		
<b>Kontaktperson:</b>	<u>Arnt Grøstad</u>		
<b>Stikkord</b> <b>(4 stk)</b>	<u>Dynamisk portal i ASP og HTML</u>		
<b>Antall sider:</b>	<b>Antall bilag:</b>	<b>Tilgjengelighet :</b> åpen	
<b>Kort beskrivelse av hovedprosjektet:</b>			
<p>Prosjektet er utviklet for at spillentusiaster skal kunne kjøpe og selge sine spill og sitt utstyr på Internett. Her finnes også spillomtaler, tips og søkefunksjoner slik at medlemmene kan finne sine favoritter. De kan også legge inn sitt favorittbilde som dukker opp hver gang de logger på. Annonsene registreres med medlemmets epost slik at de som tar i bruk portalen kan kontakte hverandre. Sidene er laget i ASP og DHTML med database i bunn.</p> <p>Sidene oppdateres i et webgrensesnitt for systemets administratorer.</p> <p>Løsningen finner du på:</p> <p>***** <a href="http://higstud5.hig.no/playstation">http://higstud5.hig.no/playstation</a> *****</p>			



# PLAYSTATION - PORTAL

  PLAYSTATION - PORTAL.com

Brukernavn:  Passord:

Bli gratis medlem i vår portal!!

Fornavn:   
Etternavn:   
Epost:   
Brukernavn:   
Passord:

Webansvarlig : Thorgrim & Kristoffer  
Telefon : 917 34 345 / 950 64 555  
E-post : [plotz@frisurf.no](mailto:plotz@frisurf.no)

## Hva skjer på Playstation-Portal?

- Les spillomtaler og nyheter!
- Søk etter spill og utstyr!
- Rediger egne annonser!
- Selg spill og utstyr!

Spillomtale

Utstyr   
Pris

Spill   
Pris

..... Søk tips .....

## FORORD

Playstation – Portal er blitt til i forbindelse med gjennomføring av hovedprosjekt ved Høgskolen i Gjøvik, våren 2001. Hovedprosjektet skal være en samlende helhetsoppgave hvor vi skal ta i bruk store deler av det vi har lært over 2 år i studiet Data- og Multimedieteknikk. Prosjektet teller 4 vektall og omfatter prosjektarbeid, rapportskrivning og programmering mot Internett.

Playstation – Portal er utviklet for Web Annonse Norge, med Stian Skoglund og Arnt Grøstad som kontaktpersoner. Den ble opprinnelig laget for å kunne ta kontakt med distributører av Playstation™ spill for videre utvikling. I prosjektet har vi lagt vekt på orden i kode og godt design.

Vi takker følgende personer:

- Arnt Grøstad som har hjulpet oss med ergonomi og kodeeksempler.
- Stian Skoglund og Web Annonse Norge som har gitt oss prosjektet.
- Øivind Kolloen, som har vært til stor hjelp i alle sammenhenger. Han var nødt til finne seg i masse spørsmål.
- Egmont – Interactive som ga oss tilgang til informasjonsserveren gjennom hele prosjektet.
- Samboere og familie som har stilt opp hele tiden.
- Medstudenter som har vært med å gi konstruktiv kritikk.

**Høgskolen i Gjøvik, 22.mai, 2001.**

---

Kristoffer Emil Amundsen

---

Thorgrim Kristiansen

# INNHALDSFORTEGNELSE

<b>1</b>	<b>INNLEDNING</b>	<b>3</b>
1.1	DEFINISJONER AV ORD OG UTTRYKK	4
1.2	OPPGAVEDEFINISJON	5
1.3	MÅLGRUPPE	6
1.4	FORMÅL	6
1.5	EGEN BAKGRUNN OG KOMPETANSE	7
1.6	ARBEIDSFORMER	7
1.7	ORGANISERING AV RAPPORTEN	8
1.8	TERMINOLOGI	9
1.9	PRAKTISK INFORMASJON OM RAPPORTEN	9
<b>2</b>	<b>KRAVSPESIFIKASJON</b>	<b>10</b>
2.1	INTRODUKSJON	10
2.1.1	BAKGRUNN	10
2.1.2	KRAV TIL SYSTEMET	10
2.1.3	SYSTEMETS OMGIVELSER	10
2.1.4	DOKUMENTETS STRUKTUR	10
2.1.4.1	INTRODUKSJON	10
2.1.4.2	BRUKERBESKRIVELSER	11
2.1.4.3	DETALJERT KRAVSPESIFIKASJON	11
2.2	BRUKERBESKRIVELSE	12
2.2.1	OMGIVELSER	12
2.2.2	SYSTEMETS MÅLGRUPPE	12
2.2.3	FUNKSJON	13
2.2.4	OPERASJON	16
2.2.5	ASPEKTER OMKRING LIVSSYKLUS	16
2.2.6	YTELSE	16
2.2.7	BEGRENSNINGER	16
2.2.8	ANTAGELSER	17
2.3	DETALJERT KRAVSPESIFIKASJON	18
2.3.1	FUNKSJONELL STRUKTUR OG TVERRELASJON	18
2.3.2	DATA RAMMEVERK	18
2.3.3	DATA INPUT	18
2.3.4	DATA OUTPUT	19
2.3.5	FILER OG LINKING	19
2.3.5.1	FILFORMATER	19
2.3.5.2	FYSISK Plassering av filer	19
2.3.5.3	LINKING MELLOM MODULER	19
2.3.6	OVERORDNEDE OPERASJONELLE SYSTEMKRAV	20
2.3.6.1	NORMAL OPERASJON	20
2.3.6.2	YTELSE	21
2.3.6.3	OPPSTART OG NEDTAKING	21
2.3.6.4	TILGJENGELIGHET	21
2.3.6.5	FEILRAPPORTERING OG SIKKERHET	21

2.3.7	FUNKSJONELLE LØSNINGER.....	22
	MEDLEM.....	22
	ADMINISTRATOR.....	22
2.3.7.1	REGISTRERING AV NYTT MEDLEM.....	23
2.3.7.2	SØKEFUNKSJONEN FOR SPILL, UTSTYR OG OMTALER.....	26
2.3.7.3	FUNKSJON SOM HENTER UT ET SKJEMA FOR UTFYLING.....	28
2.3.7.4	BESKRIVELSE AV <i>dinsideRes.asp</i> .....	30
2.3.7.5	INNLOGGING PÅ ADMINISTRATORSIDENE.....	31
2.3.7.6	OPPBYGNING AV ALLE ANDRE ADMINISTRATORSIDER.....	33
2.3.7.7	adminSalg.asp I DETALJ.....	35
2.3.8	BEGRENSNINGER.....	36
2.3.8.1	SOFTWARE ASPEKTER.....	36
2.3.8.2	DATABASE.....	36
2.3.8.3	OPERATIVSYSTEM.....	37
2.3.8.4	TOLERANSER, MARGINER OG MULIGHETER/TILFELLER.....	37
2.3.8.5	DOKUMENTASJON.....	37
2.3.8.6	SYSTEM OG PROSJEKTTESTING.....	37
2.3.8.7	KRAV TIL SUPPORT, SERVICE OG VEDLIKEHOLD.....	37
2.3.8.8	KRAV TIL UTVIDELSER.....	37
2.3.8.9	DEMOUTGIVELSE.....	37
2.3.9	AKSEPTANSEKRAV.....	37
2.3.10	PROSJEKTSTYRING, INKLUDERT KVALITETSSIKRING.....	38
<b>3</b>	<b>ANALYSE OG DESIGN.....</b>	<b>39</b>
<b>4</b>	<b>KODING OG PRODUKSJON.....</b>	<b>44</b>
4.1	ORD OG UTTRYKK I ASP.....	44
4.1.1	REQUEST.....	44
4.1.2	RESPONSE.....	44
4.1.3	RESPONSE.WRITE.....	44
4.1.4	SESSION.....	45
4.1.5	SERVER.....	45
4.1.6	JAVASCRIPT.....	45
4.1.7	HTML.....	45
4.1.8	SQL.....	45
<b>5</b>	<b>TESTING OG KVALITETSSIKRING.....</b>	<b>46</b>
<b>6</b>	<b>REALISERING / INSTALLASJON.....</b>	<b>47</b>
<b>7</b>	<b>KONKLUSJON OG DRØFTINGER.....</b>	<b>48</b>
7.1	MEDLEMSSIDENE.....	48
7.2	ADMINISTRATORSIDENE.....	49
<b>8</b>	<b>KRITIKK OG EVALUERING.....</b>	<b>50</b>
8.1	KRITIKK AV OPPGAVEN.....	51
8.2	EVALUERING AV GRUPPA.....	52
<b>9</b>	<b>ORGANISERING.....</b>	<b>53</b>
9.2	ORGANISERING OG FORDELING AV ARBEID.....	53
9.3	PROSJEKT SOM ARBEIDSFORM.....	53
<b>10</b>	<b>REFERANSER.....</b>	<b>54</b>
10.2	LITTERATUR.....	54
10.2	INTERNETT.....	54
10.3	VEDLEGG.....	54

# PROSJEKTRAPPORT

## 1 INNLEDNING

Etter å ha gjennomført et prosjekt i faget Multimedieteknikk II for DMT-Internett AS ved Arnt Grøstad og Stian Skoglund, hadde disse også et hovedprosjekt for oss. Prosjektet skulle løses på web, noe som vi var svært fornøyd med. Dette fordi vi så viktigheten av å ha gjennomført et slikt prosjekt før vi skulle ut i arbeidslivet. Oppgaven kom på banen fordi oppdragsgiverne hadde et domene som de ville ta i bruk. De ville ha oss til å lage en portal- eller forum for brukte Playstation-spill. Dette var en interessant ide som skulle utvides til et mer helhetlig prosjekt. Vi satte oss ned sammen og definerte de første rammene for prosjektet som fikk navnet Playstation-Portal. Denne skulle inneholde kjøp og salg av både spill og utstyr tilhørende Playstation. I tillegg skulle medlemmer få mulighet til å lese omtaler, se bilder og hente ut tips som andre medlemmer hadde lagt inn.

Vår oppgave ble å gjennomføre prosjektet med det for øye å gi nødvendig funksjonalitet og dynamikk til at administratorer kunne oppdatere databasetabellene som lå i bunn.

I tillegg ville vi at medlemmene skulle kunne finne riktig informasjon og ta kontakt med andre medlemmer på en enkel måte. Prosjektet skulle også være et eksempel for andre som ville lage dynamiske sider som kunne oppdateres på Internett. Annen funksjonalitet kunne tas med, men dette var ikke et krav. Web Annonse Norge som er offisiell oppdragsgiver ga oss ideen Playstation – Portal med forholdsvis frie tøyler til å definere design og innhold. I denne rapporten kan du lese om prosessen bak.

Rapporten er bygd opp på følgende måte:

I hovedrapporten ligger kravspesifikasjon og en generell beskrivelse av prosjektet. Dessuten kan du lese om design og enkelte kodesekvenser som vi har tatt tak i. Som vedlegg finner du kildekode, gantt-skjema/framdriftsplan, statusrapporter, møtereferater, loggbok og timelister.

## 1.1 DEFINISJONER AV ORD OG UTTRYKK

Access	Database utviklingsverktøy fra Microsoft
ASP	Active Server Pages
Browser	Nettleser, program/software som leser Internett-sider
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
DPI	Dot Pr. Inches/Punkter per tomme
E-post	Elektronisk post sendt over Internett
Form	Skjema; et element i dokument strukturen i en HTML-fil til innsamling av informasjon og valg fra brukeren
GIF	Graphic Interchange Format - filformat for bilder etc.
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
Hyperlink	En elektronisk referanse fra et HTML dokument til et annet dokument eller til en bestemt del av et dokument.
IIS	Internet Information Server
JavaScript	Scriptspråk hvis kode skrives i HTML dokumentet og eksekveres i brukerens browser
JPEG	Joint Picture Experts Group - filformat for bilder etc.
Klient	I Internett sammenheng: browseren på maskinen tilknyttet Internett som sender en forespørsel om en HTML-fil.
Microsoft Windows NT	Operativsystem
Playstation – Portal	Web løsningen/applikasjonen beskrevet i dette dokumentet, også noen steder kalt "systemet"
Portal	Webside som inneholder sammenfattet informasjon om et emne. Møtested for personer som søker informasjon
PP	Web løsningen/applikasjonen beskrevet i dette dokumentet
PWS	Microsoft Personal Web Server
Script	Et program hvis kode ikke er ferdig kompilert maskinkode, men må tolkes og oversettes til maskinkode ved prosessering
SMTP	Standard Mail Transfer Protocol
Stylesheet	Et dokument som HTML dokumentene er linket til for å oppnå en bestemt tekst stil og/eller layout. Det samme som CSS
Upload	Opplasting av filer fra klient til server
URL	Uniform Resource Locator. Adressen til et dokument på Internett
VBScript	Scriptspråk som på lik linje med JavaScript kan eksekveres i brukerens browser. I forbindelse med ASP benyttet til å kjøre script på serveren
Webserver ved	Software som gjør det mulig for andre tilknyttet Internett, Forespørsler til serveren, å kunne få tilgang til filer residerende på datamaskinen serveren er installert på
Webside	Nettsted/avgrenset område på Internett



## **1.2 OPPGAVEDEFINISJON**

Av oppdragsgiver fikk vi en oppgavetekst som ga oss stort spillerom innenfor det å definere oppgavens størrelse. Innholdet var i rammer stort sett klart etter samtaler i startfasen. Prosjektet ble gitt for å mulig kunne interessere Playstation- aktører i Norge til samarbeid og salg av løsningen.

### **OPPGAVETEKST:**

Portalen som tenkes opprettet, vil bli realisert med ASP- programmering mot en Access-database. Dette vil bli utført på den måten at du på et web- brukergrensesnitt kan legge inn all informasjon som deretter blir lagret automatisk i databasen. Dette vil jobbe som et eget program på en server og holde orden på all informasjon automatisk.

Vi vil bruke ASP, SQL og DHTML som utviklingsspråk. Portalen skal være et bidrag til å hjelpe spille- glade personer til å få kjøpt og solgt spill og utstyr på en enkel måte. Alle kan bli medlemmer, og det skal ikke koste noe.

### **MÅLSETTING:**

Når prosjektet er avsluttet skal det foreligge en fungerende dynamisk webportal, med tilhørende databaser og eksempel arkiv. Portalen skal også kunne være et utgangspunkt for et konkurransedyktig nettsted i forhold til lignende sider.

### **VERKTØY/PROGRAMMERING:**

Følgende teknologi ble brukt under prosessen:

Microsoft Access

MS InterDev/Notepad/EditPlus

ASP

JavaScript

CSS

Adobe Photoshop

## **FORUTSETNINGER/BEGRENSNINGER**

Det er forventet at oppdragsgiver har gjort undersøkelser på viktigheten og nødvendigheten av en slik løsning. Grappa bestemmer design, omfang og funksjonalitet, med innspill fra oppdragsgiver. Med bakgrunn i dette er følgende forutsetninger satt for oppgaven:

Webserver plattform: Windows NT med IIS med mulighet for ASP som serverspråk  
Administratorer har en kort opplæringsperiode med gjennomgang av funksjonalitet  
Access database

Løsningen er utelukkende for web

Medlemmer kan legge inn så mange spill de vil

Medlemmer kan kommunisere ved hjelp av e-post

Medlemmer må ha brukernavn og passord for å få tilgang til portalen

Løsningen er optimalisert for IE 5.0 og NC 4.7

Både administratorer og medlemmer kjenner generell bruk av Internett

Skjermopløsning må være minimum 800x600

Skjerm må kunne vise minimum 256 farger

Hastighet på modem (forbindelse) må være minimum 56 Kb/s

Nedlastingstid på sidene skal værers maksimum 25 sek

## **1.3 MÅLGRUPPE**

For rapporten:

Lesere av rapporten er veileder og sensorer som skal vurdere prosjektet

Data- interesserte og nybegynnere som vil ha hjelp til egne prosjekt

Oppdragsgivere som skal kunne se hvordan prosessen har gått

For portalen:

Medlemmer - De som logger seg på og benytter portalen som en servicetjeneste.

Administratorer – de som vedlikeholder portalen, med full tilgang til all informasjon.

## **1.4 FORMÅL**

En del av formålet med oppgaven er å fremlegge et resultat som kan være med på å styrke en fremtidig arbeidsgivers inntrykk om at vi behersker de teknologier som er fremtredene i dagens utvikling av websider. Vi vil også presentere et produkt som oppdragsgiver kan bruke som plattform for en portal på Internett. Selve innholdet i oppgaven er ingen av oss godt kjent med, men vi syntes det hørtes interessant ut. Da oppdragsgiver presenterte ideen fikk vi vite at det var en oppgave som ga programmeringsutfordringer som vi ville møte i arbeidslivet. Et webprosjekt var dessuten noe vi ikke hadde arbeidet med før, noe som ville gi oss en bredere forståelse av IT-miljøet i dag.

## 1.5 EGEN BAKGRUNN OG KOMPETANSE

Vi hadde lite erfaring med programmering på klient/server- nivå. Annen erfaring har vi fått gjennom de to årene på Data- og Multimedieteknikk- studiet. Kristoffer har i tillegg hatt programmeringsfaget IN105 på universitetet i Oslo, og han driver sitt eget datafirma med salg av hardware og komponenter. Thorgrim hadde ingen dataerfaring før de to årene ved HiG, og besøkte Internett for første gang i 1999.

Dette gjorde at vi måtte ha en høy læringskurve for å kunne gjennomføre prosjektet. Gjennom prosessen har vi tilegnet oss kunnskap i mange felter, blant annet ASP(VB-script), Access-databaser, SQL-spørringer, HTML, JavaScript, CSS, Photoshop, layout, ergonomi, prosjektarbeid, rapportskrivning og planlegging.

## 1.6 ARBEIDSFORMER

På grunn av at vi var kun to personer, måtte vi jobbe mye sammen og dele ansvaret for at arbeidet ble gjort. En naturlig oppdeling var å ta hovedansvar for hver sin hoveddel. Kristoffer fikk ansvaret for administratorsidene, mens Thorgrim lagde medlemssidene. Dette gjorde at begge fikk en helhetlig arbeidsprosess med programmering, layout og databasebehandling. Dette var for å sikre at vi lærte mest mulig om flest mulige deler av prosjektet.

Prosjektarbeidet startet med en planleggingsfase som inneholdt mye administrativt i form av arbeidsfordeling, målspesifikasjon og oppgavebeskrivelse. Oppgaven delte vi opp i hoveddeler, som også er spesifisert i gantt-skjemaet (se vedlegg B).

Hoveddelene ble inndelt slik:

Planlegging

Ergonomi

Design

Databaser

Pseudokoding av ulike elementer

ASP Programmering

HTML Programmering

Kvalitetskontroll

Testing

Rapporter

Fremføring av prosjekt

Gruppens arbeidsformer har vært møter med informering av den andre parten, og planlegging av ukeoppgaver. Selve arbeidet har foregått individuelt med arbeid hjemme og på skolen. Møter med oppdragsgiver har for det meste foregått over telefon og e-post, men vi har hatt to møter i Hamar med Arnt Grøstad. Veileder har vi hatt hyppig kontakt med under prosessen. Dette har vært fastsatte møter og korte spørsmålssekvenser (se også møtereferater, vedlegg D)

## 1.7 ORGANISERING AV RAPPORTEN

**Kapittel 1** inneholder en redegjørelse om hva som skal produseres og hvorfor. I dette inngår en kort beskrivelse av organisasjonen som skal benytte løsningen. Generelle, overfladiske krav til løsningen, samt en definisjon over spesielle ord, uttrykk og forkortelser som benyttes i kravspesifikasjonen, er også beskrevet. Dette for å skape et bilde av de krav som stilles til løsningen og hvordan den skal brukes. Definisjoner på ord og uttrykk som er brukt i spesifikasjonen, finnes også her.

**Kapittel 2** beskriver kravspesifikasjonen. Første del er bakgrunnen for prosjektet, hvilke begrensninger tidsplanen setter, og kostnader i forbindelse med prosjektet. Videre sier den noe om systemets omgivelser og dokumentets struktur.

Del to i kravspesifikasjonen inneholder mer detaljert beskrivelse av systemet og dets funksjoner og brukere. Den sier også noe om hvordan systemet skal fungere og livssyklusen til systemet. I tillegg sier den noe om ytelser, forutsetninger og begrensninger.

I del tre blir funksjoner og struktur spesifisert. Blant annet sies det noe om hvilke data som går hvor, filformater og fysisk plassering av filer. Overordnede operasjonelle systemkrav som hvordan systemet skal fungere under normal operasjon, ved feilsituasjoner, systemets ytelse, oppstart/nedtaking og tilgjengelighet er også beskrevet her. Funksjonelle krav og begrensninger med henhold til programvare, språk, kommunikasjonsstandarder og database er spesifisert. I tillegg beskrives hvilket operativsystem som kan brukes, aspekter omkring livssyklus, aspekter omkring installasjon, utgivelser underveis og akseptansekrav. Til slutt sier denne delen noe om prosjektstyring og kvalitetssikring.

**Kapittel 3** inneholder beskrivelse av design og oppbygning av kodesekvenser. Her redegjør vi for metoder og prosesser bak oppsettet av selve portalen. Kapitlet skal forklare deg hvordan og hvorfor vi har gjort det vi har gjort. Alt er ikke forklart, men vi har tatt opp det som er viktig for å beskrive vårt arbeid og vårt resultat.

**Kapittel 4** beskriver hvilke programmerings- og markeringsspråk som er brukt. I tillegg blir det også redegjort for hvilke verktøy som er brukt ved behandling av bilder og tekst.

**Kapittel 5** sier noe om testing og kvalitetssikring av løsningen.

**Kapittel 6** sier noe om realisering og installasjon.

**Kapittel 7** tar for seg de forskjellige delene av prosjektet slik de var planlagt og slik de har blitt, samt forslag til hva som kunne ha vært gjort annerledes. I tillegg tar kapitlet for seg en helhetlig konklusjon av løsningen.

**Kapittel 8** tar for seg kritikk av oppgaven.

**Kapittel 9** inneholder evaluering av gruppa.

## 1.8 TERMINOLOGI

Bruker	Medlemmet eller administrator ettersom hvilke sider man er på
Han	Medlemmet
Hun	Administrator
Vi	Medlemmene i gruppa
Du	Leseren av rapporten
Veileder	Øivind Kolloen
Arnt	Kontaktperson hos oppdragsgiver
PP	Playstation-Portal. Navnet på portalen
IE	Internett Explorer
NC	Netscape Communicator
Oppdragsgiver	Web Annonse Norge

## 1.9 PRAKTISK INFORMASJON OM RAPPORTEN

Vi valgte følgende standard fonter, størrelser og stiler:

Hovedoverskrift:	Arial 16 pkt. fet.
Overskrift:	Arial 14 pkt. fet.
Underoverskrifter:	Arial 12 pkt. fet.
Kildekode:	Arial 10 pkt, kursiv, blå.
Kommentering:	Arial 10 pkt, fet.
Bildetekster:	Arial 10 pkt.

Alle henvisninger skal stå i parentes.

Vi har valgt å legge teksten i én spalte og bruke topptekst og bunntekst. Rapporten består av flere dokumenter. Toppteksten forteller leseren hvilket dokumentet han er i og bunnteksten forteller hvilken side. Figur angis med figur, tabell angis med tabell og kildekode angis med kildekode.

## 2 KRAVSPESIFIKASJON

Kravspesifikasjonens oppsett er basert på en oversettelse utført av Frode Haug av "The Starts Purchasers Handbook", kap. 4 og appendix B. Vi har arbeidet med kravspesifikasjonen gjennom hele prosjektets varighet. De to første delene har stort sett ikke blitt endret siden 12. februar, som var den første innleveringsfristen. Del tre har vi videreutviklet gjennom hele prosjektperioden, den ble ferdig like før prosjektets innlevering.

Plan for gjennomføring av hovedprosjektet framgår av framdriftsplanen (se vedlegg B). Den grafiske framstillingen av prosessen og hvordan den har blitt gjennomført, vises i gantt-skjemaet (se vedlegg B).

### 2.1 INTRODUKSJON

#### 2.1.1 BAKGRUNN

Oppdragsgiver er Web Annonse Norge, med Arnt Grøstad og Stian Skoglund som kontaktpersoner. Disse står for drift av ulike internettsider, gjennomføring av komplette webprosjekter og arbeider i bedrifter på Hamar (Ipares /Apropos Internett) og Lillehammer (Web Annonse Norge). Oppdraget vi har fått går ut på å utvikle en demoversjon av en spilleportal på Internett der folk kan bli medlemmer, og deretter kjøpe og selge spill og spillartikler. Medlemmene skal kunne legge inn koder og triks til spill. Det vil dessuten bli opprettet sider for administratorer som skal kunne oppdatere portalen automatisk over Internett. Prosjektet var i utgangspunktet ment for salg til et større firma innen webutvikling og e-tjenester, men det er også vurdert å legge det endelige produktet til et eget domene.

#### 2.1.2 KRAV TIL SYSTEMET

Det er helt klare begrensninger når det gjelder tidsforbruket på et slikt prosjekt. Gruppen består av to personer, og skal i tillegg til å utføre oppgaven lære nye programmer og programmeringsteknikker. 4 vekttall tilsvarer ca. 20 timer per person i uka, og dette er også en begrensning. Kostnadene er fra oppdragsgiver ikke definert, men prosjektet baserer seg på utgifter som tomme CDer, reiser, fargepatroner til printere o.l.

#### 2.1.3 SYSTEMETS OMGIVELSER

Vi arbeider på tre forskjellige steder; Hjemme hos Kristoffer og hos Thorgrim samt på skolens datalab. A-209. Hjemme benyttes verktøyet Microsoft Visual InterDev med Personal Web Server og EditPlus, mens på skolen benyttes Notepad, EditPlus og Microsoft Visual InterDev med Internett Information Server.

#### 2.1.4 DOKUMENTETS STRUKTUR

Kravspesifikasjonen er bygd opp på følgende måte:

##### 2.1.4.1 INTRODUKSJON

Under dette punktet forklarer vi bakgrunnen for prosjektet, rammer for tid og kostnader. I tillegg beskrives forholdene vi arbeider under, programvare og dokumentinformasjon.

**2.1.4.2 BRUKERBESKRIVELSER**

Denne delen tar for seg systemet og dets brukere. Vi tar for oss hvordan systemet og funksjonene skal fungere, hvor lenge systemet skal virke og hva slags ytelse og forutsetninger som settes for det. I tillegg skal det defineres brukernes omgivelser og generell informasjon om hvem brukerne er.

**2.1.4.3 DETALJERT KRAVSPESIFIKASJON**

Her defineres operativsystem, filformater, dataforbindelser, databaser, systemkrav og plassering av filer. Andre aspekter er funksjonelle krav under vanlig drift og feilsituasjoner, samt installering og drift av systemet. Arbeidsrammer med henhold til programmer, verktøy og språk vil forklares, og i denne delen spesifiseres funksjoner og struktur. Blant annet sies det noe om hvilke data som går hvor, filformater og fysisk plassering av filer. Funksjonelle krav og begrensninger mht. software, språk og kommunikasjonsstandarder er også spesifisert her. Denne delen er endret og modifisert i løpet av prosjektet og vil avvike fra den første innleverte kravspesifikasjonen. Prosjektet var en læringsprosess som førte til at vi fant bedre, mer effektive måter å bygge opp sider på. Derfor har spesifiseringen også endret seg.

## 2.2 BRUKERBESKRIVELSE

### 2.2.1 OMGIVELSER

Systemet skal fungere på en 5.0 Internett Explorer nettleser (og nyere), og Netscape 4.7. Sidene skal ligge på en Windows NT 4.0 eller Windows 2000 server med IIS 4.0. Sidene er designet for skjermer med minimum 256 farger. Minimumskrav for modem er 56 Kbps.

### 2.2.2 SYSTEMETS MÅLGRUPPE

Målgruppen er todelt.

Medlemmer - de som logger seg på og benytter portalen som en servicetjeneste.

Administratorer – de som vedlikeholder portalen, med full tilgang til all informasjon.

Det forutsettes at målgruppen "Medlemmer" har grunnleggende kunnskaper om data og bruk av Internett. En annen forutsetning er at de har en e-post adresse, og er kjent med å sende og motta e-post. Da det ikke formidles noe som kan være krenkende eller støtende av seksuell eller voldelig art, er det ingen aldersgrense for å melde seg inn.

Administratorer bør også ha kjennskap til Internett og E-post. Administrator trenger ikke å ha kjennskap til databaser, HTML eller ASP, da all oppdatering og endring gjøres ved å fylle ut skjemaer på egne administrator sider. En administrator kan legge til en ny administrator i databasen. Det forutsettes en kort opplæringsperiode for eventuelle administratorer som ikke er kjent med systemet. Denne målgruppen må ha tilgang til en Internett-tilkobling da alt vedlikehold gjøres fra en web-side. Sammensatt i punkter ser målanalysen slik ut:

- Han som vil bruke "Playstation-Portal" er fortrinnsvis en person som eier Playstation eller som hyppig spiller Playstation-spill
- Han er av begge kjønn, i en aldersgruppe hovedsakelig fra 10 til 40 år
- Han vil besøke siden for å finne informasjon
- Hovedsakelig har han en Internett- tilkobling fra 56K modem til ISDN (opptil 128 Kb/sekund)
- Han har allmen datakunnskap
- Han og hun forstår norsk
- Hun har god allmen datakunnskap
- Han og hun kjenner Internett-teknologi og bruk av skjemaer og knapper på web.
- Han og hun kjenner noe til Playstation-artikler



### 2.2.3 FUNKSJON

Funksjonene er delt opp i to hovedmoduler, henholdsvis medlem og administrator. Disse har vi splittet og forklart hver for seg:

#### Medlem:

Alle sidene utenom *index.asp* har en funksjonsfil som inkluderes på alle sider. Denne inneholder alle funksjonene som brukes i PP. Funksjonene blir kalt på sidene avhengig av hvilke if-tester som slår inn. *dinsideRes.asp* sender data til seg selv og mottar data fra de andre sidene i tillegg.

- *index.asp*  
Tar seg av pålogging og registrering av eksisterende og nye medlemmer. Alle som vil ha tilgang må skaffe seg tilgang på denne siden.
- *dinside.asp*  
Hovedsiden med informasjon om personlige annonser, opplasting av personlig bilde og generelle søkefunksjoner. Siden viser melding om endringer i personlige annonser.
- *dinsideRes.asp*  
Resultatside som lister ut de resultatene som det blir søkt på. Denne siden står for over halvparten av resultatvisning for medlemmet. Siden viser spill- og utstyrsannonser, tips og omtaler.

- *selge.asp*

Side som viser skjema for utfylling av ulike annonser som spill, utstyr og tips.

#### *annonse.asp*

Tar for seg visning av komplette annonser, og her kan han endre eller slette sine innlagte annonser.

**Administrator:**

Alle sidene er bygget opp på samme måte unntatt *index.asp* som på grunn av at administratoren ikke har logget seg inn ennå. Det finnes også en del sider som hun ikke ser. Disse sidene inkluderes i eksempelsidene under. Dette være seg *adminTopp.asp* som tegner opp logo, oppretter databaseforbindelse samt lager meny og indirekte dato for visning. Datoen er i seg selv en fil som inkluderes, *adminDato.asp*. *adminTopp.asp* inkluderer også siden *session.asp* som sørger for å kontrollere at sesjonen er gyldig. På bunnen av sidene inkluderes *adminBunn.asp*, som lukker åpne objekter, samt tegner opp en logg-ut-knapp. Denne sender resultatet til *adminBunnRes.asp* som terminerer sesjonen og logger ut en administrator. Sidene som hun får se er som følger:

- *index.asp*  
Her kan administrator logge seg på systemet. Siden sørger for å opprette en sesjon og sende administrator videre til *adminHoved.asp* hvis informasjonen ved innloggingen blir godkjent. Hun kan også gå til medlemssidene via en link.
- *adminHoved.asp*  
Hovedsiden som viser henne aktuell informasjon hentet fra databasen.
- *adminNyttSpill.asp*  
Her kan hun legge inn nye spill til databasen.
- *adminSlettSpill.asp*  
Her kan hun slette spill fra databasen, for eksempel hvis et spill er feilaktig lagt inn.
- *adminOmtale.asp*  
Her kan hun legge til nye spillomtaler, redigere gamle, eller slette de hun ikke vil ha med i systemet lenger.
- *adminMedlem.asp*  
Her vises en oversikt over alle registrerte medlemmer. Hun kan slette medlemmer herfra, hvis noen for eksempel har oppført seg uetisk, skrevet støtende innhold i annonsene sine el. Hun kan også finne en link som henviser til sending av e-post til et enkelt medlem.
- *adminSalg.asp*  
Her kan administrator redigere eller slette salgsannonser, både for spill og utstyr.
- *adminTipsOgTriks.asp*

Samme som for *adminSalg.asp*, men for tips, juksekode og spillgjennomganger.

- *adminNyAdmin.asp*

Her kan administrator registrere en ny administrator. Hun vil trenge å skrive inn administratortnavnet sitt og tilhørende passord for å opprette en ny administrator. Dette for å ha en ekstra sikkerhet i forhold til at fremmede skal kunne bli administratorer. Et eksempel kan være hvis administrator går for å hente seg en kopp kaffe, men hun glemmer å logge seg ut. Noen kan dermed ikke gå inn i systemet å gjøre seg selv til administrator. Hvis det går lang tid mellom hendelser på systemet vil systemet sørge for at sesjonen blir ugyldig. Administrator må da uansett logge seg inn på nytt. Siden viser også en liste over registrerte administratorer, dog uten å vise deres passord.

- *adminSlettAdmin.asp*

Her kan hun (administratoren) slette en administrator ut i fra en rullegardinmeny med navn. Dette er kanskje litt enkelt, men ingen kan ødelegge systemet ved at ingen kan administrere det. En løsning på dette problemet (hvis det ikke er noen registrerte administratorer på systemet) kunne være å ha en selvstendig side hvor man kunne opprette én administrator hvis ingen andre var registrert. Denne siden måtte spørre etter et passord som kunne utleveres av utviklerne (oss). Eventuelt kunne en person som hadde kjennskap til Microsoft Access™ gå inn i databasen manuelt, og legge til en ny administrator. Dette er også en svakhet da databasen ikke er kryptert på noe vis, men ligger som en vanlig .mdb (Microsoft DataBase) fil. Dette kunne også vært en forbedring i fremtidige versjoner av systemet.

- *adminLinker.asp*

Her kan administrator legge til eller fjerne linker som synes på medlemssiden *dinside.asp*. Navnet på linken kan ikke være mer enn 15 tegn langt. Dette av kosmetiske årsaker.

#### **2.2.4 OPERASJON**

PP er tenkt som en service til personer som spiller Playstation 1 og 2.

Ved å bli medlem kan man legge ut annonser på spill og utstyr som skal selges, samt legge ut informasjon om spillkoder, triks og spillgjennomganger. All denne informasjonen lagres i en database, og et medlem kan dermed få tilgang til koder, triks, omtaler og annonser som andre medlemmer har lagt inn. Systemet skal inneholde informasjon og feilmeldinger som gir medlemmene forklarende meldinger om hva som er gjort og/eller eventuelle ting som må gjøres.

Det endelige produktet er forespeilet en levetid som er ubegrenset så lenge det er en tjeneste som brukere vil ha og benytter seg av. Det er ikke fastsatt noen ansvarlige til å drifte denne tjenesten, men PP er også en demoversjon som kan brukes til opplæring og oppbygging av andre liknende websteder. Den eventuelle serveren som vil ta seg av tjenesten mener vi bør ha en oppetid på tilnærmet 100%. De få gangene den er nede beregnes til eventuell strømstans, overkapasitet og menneskelige feil.

#### **2.2.5 ASPEKTER OMKRING LIVSSYKLUS**

Systemet må driftes og vedlikeholdes av eventuelle kunder som oppdragsgiver kontakter. Utvidelser gjøres ut fra oppdragsgivers/kundens behov. Løsningen kan brukes på hvilke som helst maskiner nevnt under punkt 2.2.1, men på grunn av at ASP er brukt må dette tas med i betraktning hvis PP skal brukes på en annen plattform. Under utvikling av prototypen vil testing og kontroll av nye funksjoner foregå i samarbeid med kontaktpersonene og veileder.

#### **2.2.6 YTELSE**

Vi ønsker max. nedlastingstid av sider på 25 sek pr. side for medlemmene.

#### **2.2.7 BEGRENSNINGER**

Oppdragsgiver har ikke lagt noen spesielle software- eller hardware begrensninger på gruppen, men vi har fått tilgang til et utviklingsverktøy som brukes av mange IT-selskaper. Microsoft InterDev er derfor installert hjemme og på skolen, og det er brukt over en periode. EditPlus er en annen favoritt som er også har blitt brukt i stor grad.

### 2.2.8 ANTAGELSER

Det er ikke foretatt noen markedsundersøkelse eller analyse i forkant av prosjektet, men det forutsettes at oppdragsgiver har foretatt nødvendige undersøkelser for å klarlegge markedet til denne portalen. Utfra dette har vi satt disse forutsetningene for Playstation-Portal:

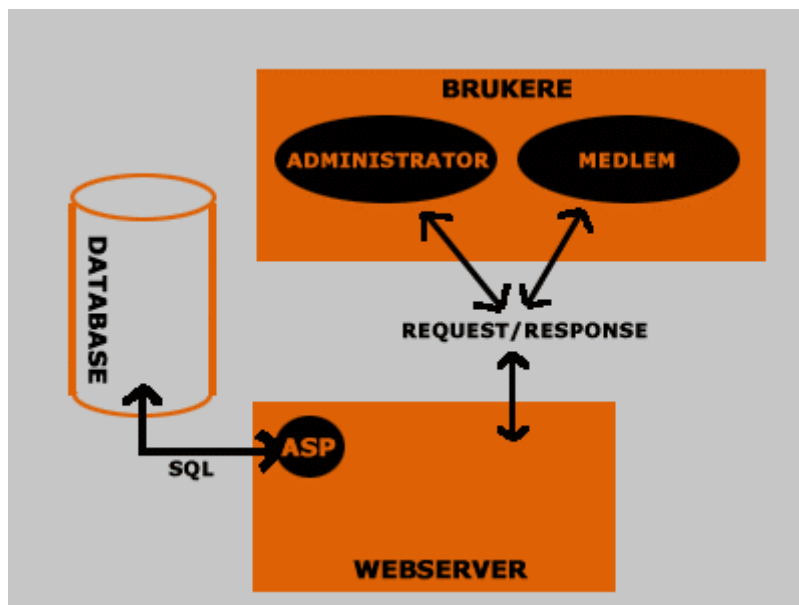
- Web server plattform: Windows NT med IIS (for ASP) tilknyttet nettet, eller Windows98/2000 med Frontpage Extension Server
- Microsoft Access database
- Løsningen er bare laget for Internett
- Påmelding blir automatisk registrert i databasen
- Han som ikke er medlem vil ikke få tilgang til siden
- Hun kan overstyre alle funksjoner på egne sider
- Hun skal kunne oppdatere alt fra et webgrensesnitt (Spesielt for videre utvikling)
- PP krever minimum en 5.0 IE browser eller en 4.7 NC browser
- Han og hun kjenner Internett og bruk av e-post
- Han har egen e-postadresse
- Skjermopløsning 800 x 600 piksler
- Skjerm med minimum 256 farger.
- Minimum 15" skjerm.
- Hastighet på modem (forbindelse til Internett): minimum 56 Kb/s.

## 2.3 DETALJERT KRAVSPESIFIKASJON

### 2.3.1 FUNKSJONELL STRUKTUR OG TVERRELASJON

For nærmere beskrivelse av funksjonell struktur og sideoppbygging se beskrivelse under punkt 2.2.3.

### 2.3.2 DATA RAMMEVERK



Figur 2.3.2.1: Hvordan PP fungerer. Maskinvaren som ligger bak

### 2.3.3 DATA INPUT

Input fra medlemmer vil gjøres fra webgrensesnitt ved hjelp av forms og den teknologiens attributter. Dataoverføring gjøres på standard måte, HTTP og URL-tilkobling over Internett. For eventuelle administratorer vil det være mulig å få tilgang til selve webserveren over Internett med FTP, eller gå inn lokalt, noe som gjøres fra selve operativsystemet på den enkelte server. Request er et objekt i ASP som gjør at serveren kan ta imot input fra målgruppene og behandle data. Dette skjer på et HTML eller ASP dokument. Response utgjør output til målgruppen. Dette objektet styrer hva som skal sendes tilbake til klienten basert på tidligere input. Objektet sender han til en ny side.

### 2.3.4 DATA OUTPUT

Output er data som serveren sender. Dette baserer seg på input fra målgruppen, og spørringer som gjøres mot eventuelle databasetabeller. Output vises på et HTML eller ASP dokument som browseren til klienten tar imot. Metoden for å gjøre dette blir i stor grad utført ved hjelp av response-objektet. Ved enkelte tilfeller vil det også kunne vises statiske sider. Da trenger ikke response-objektet være aktivt.

**REQUEST** innebærer forespørsel om et HTML/ASP- dokument eller en streng som parameter.

**RESPONSE** kan være å gå til et HTML/ASP- dokument eller skrive noe til dokumentet.

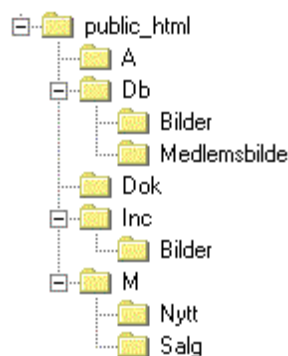
### 2.3.5 FILER OG LINKING

#### 2.3.5.1 FILFORMATER

HTML/HTM, ASP, INC, CSS, MDB, GIF, JPG,

#### 2.3.5.2 FYSISK PLASSERING AV FILER

Under følger en skisse på hvordan katalogstrukturen er bygd opp. Dette valget er gjort for å ha en god oversikt over hvor filer ligger. Navnsettingen av filene har gjort det oversiktlig innad i katalogene. Vi har valgt å lage fire hovedkataloger, disse gir et naturlig skille for strukturen. Stylesheet er ikke plassert i egen katalog, men er innlemmet i katalogen **Inc**.



Figur 2.3.5.2.1: Katalogstruktur

#### 2.3.5.3 LINKING MELLOM MODULER

For å lett kunne flytte systemet til en annen server har vi valgt å bare bruke relative adresser mellom filene/sidene.

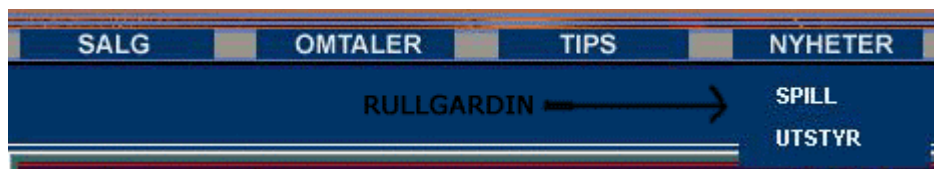
## 2.3.6 OVERORDNEDE OPERASJONELLE SYSTEMKRAV

### 2.3.6.1 NORMAL OPERASJON

Systemet vil operere på to ulike plan:

- Medlemmer (Han)
- Administrator (Hun)

Medlemmet skal kunne navigere seg rundt ved hjelp av den horisontale menyen som er innlemmet i logoen. Dette er gjort på den måten at bildet inneholder referanser som gjør at musa oppfatter områdene i bildet som linker (bildet bruker mapping). Denne menyen inneholder et JavaScript som gjør at når musa er over en knapp så vil det komme ned en rullegardinliste med linker til andre valg innenfor den kategorien. Dette skal være en brukervennlig måte som er effektiv, synlig og som sparer plass. Se figur 2.3.6.1.1 for nærmere forklaring.



Figur 2.3.6.1.1. Ved musepekeren over "NYHETER" vil denne rullegardinen komme ned.

Kommunikasjon med databasen via ASP-delen blir styrt med funksjoner som får tilsendt informasjon fra forms og ved sending av parametre på hyperlinker. Annen navigasjon vil foregå ved standard hyperlinker, eller ved hjelp av response-objektets redirect metode. Siden er oppdelt slik at venstre del er hans mulighet til å legge inn, søke og hente ut informasjon fra databasen. Denne er fast på alle sider, og vil bli en personlig menybar som vil dekke de behov som vi har satt for målgruppen for medlemmer.

Administrator har mulighet til å oppdatere enhver tabell i databasen. Vi så det som viktigst at all informasjon som har med kjøp og salg kunne forandres på. De designmessige oppdateringene har ikke vært en del av oppgavedefinisjonen, men det er laget funksjoner for å legge inn linker og for å skrive dagens tekst som blir liggende på hovedsiden (sistnevnte er under utarbeiding). Funksjoner som er utviklet for han og henne er brukernavn og passord beskyttet.



### 2.3.6.2 YTELSE

Vi ønsker at responstiden skal være maks 25 sek. Bortsett fra bilder er det ikke innhold på sidene som vil bli veldig store, og gruppa mener at det skal være brukervennlig også for modembbrukere. Med 7 KB i sekundet bør ikke siden overstige ( 7KB \* 25 sek) 175 KB. Dette ble sjekket på Betaversjonen som forelå ved utgangen av uke 14. Oppbyggingen av sidene er i dag totalt forskjellig, og tester på ytelse er dermed gjort i uke 20. Ved innhenting og opplasting av informasjon fra databaser vil navigering foregå ved hjelp av form-knapper og hyperlinker med medsendte parametre. For brukere av Internett Explorer vil det brukes en hover- funksjon på linkene. Dette gjør at linkene forandrer farge når musa er over. Dette er ikke støttet i Netscape, men bortsett fra det vil utseende på linker være de samme. Vi har ikke satt noen krav til antall samtidige brukere, men koden er forsøkt optimalisert for sider med ASP-interaktivitet. Prosjektet skal enkelt kunne flyttes til andre webservere som støtter ASP. Responstiden måles fra start nedlasting til slutt nedlasting.

### 2.3.6.3 OPPSTART OG NEDTAKING

Dette prosjektet bygger på en levetid som avhenger av markedet. Når det er installert er det meningen at webområdet kan brukes helt til oppdragsgiver, kunder osv. synes det ikke gir gode nok resultater. Systemet kan i tillegg brukes som en mal for andre liknende sider, og i slike tilfeller er levetiden avhengig av opplæringstiden.

Det er beregnet at Playstation-Portal skal kunne kjøres på alle servere som støtter ASP og som bruker Microsoft Access 97. Vi mener at det vil være tilfeller der systemet går ned og situasjoner som kan utgjøre dette kan være overbelastning, strømstans, avslutning av webområder osv. Vi har valgt å ikke sette tall på oppetiden, da dette vil avhenge av hardware og belastning.

### 2.3.6.4 TILGJENGELIGHET

Så lenge kunden vil ha det i drift vil systemet kunne brukes.

### 2.3.6.5 FEILRAPPORTERING OG SIKKERHET

Feil som oppstår ved brukerhendelser vil tas hånd om av ASP og JavaScript. Disse språkene har mulighet for å gi tilbakemeldinger på feil inntasting, mangel på tilgjengelighet og feil i databasesøk. Brukeren vil få godkjenning eller feilmelding. Begge målgruppene for PP vil bli laget ved hjelp av en sesjonsvariabel som gjør det svært sikkert mot inntrengere og hackere. Sesjonstiden er ikke fastsatt, men avhenger av oppsett på hver enkelt server som bruker PP. Ytterligere sikkerhet er ivaretatt ved at han og hun må logge seg på for å benytte seg av sin modul. Dette gjør for det første at upassende informasjon blir knyttet til en person, slik at dette kan spores og fjernet fra nettstedet. For det andre fører dette til at innhold i databaser endres av personer som har et reelt formål.

### **2.3.7 FUNKSJONELLE LØSNINGER**

Playstation-Portal kan deles inn i følgende deler:

#### **MEDLEM**

Innlogging eller registrering

Generell informasjon

Søk på spill, utstyr og omtaler

Linker til relaterte søk

Opplasting av personlig bilde

Oppretting av egne annonser

Redigering av egne annonser

Se de nyeste annonsene til enhver tid

E-post som gir medlemmet mulighet til å kontakte et annet medlem

Utlogging

#### **ADMINISTRATOR**

Innlogging

Informasjon om antall medlemmer og annonser som finnes i databasen

Legge inn omtaler, spill, utstyr, dagens tekst, linker, bilder og administratorer

Endre medlemmers annonser (for eksempel fjerne stygge ord og uttrykk)

Sletting av samtlige poster i databasen (en og en)

Generering av e-post til samtlige medlemmer

Utlogging

Under følger forklaringer og eksempler på kode brukt i prosjektet. Fullstendig kildekode til sider vi viser til i dette kapittelet (kalt eksempelsider) finnes som vedlegg A. Medlemssidene vil bli tatt for seg først, mens administratorsidene vil bli omtalt etterpå. Kodebiter som er like på administratorsidene som de var på medlemssidene vil ikke bli forklart ytterligere, og heller ikke tatt med i oppsummeringen. Derfor er administratordelen her betydelig kortere enn medlemsdelen.

### 2.3.7.1 REGISTRERING AV NYTT MEDLEM

PP kan kun tas i bruk ved å logge seg inn. Dette har vi gjort for å kunne kontrollere innhold og knytte hendelser og innhold det til et navn(e-postadresse). Dessuten er det fint å kunne se hvor mange medlemmer som tar i bruk PP. Siden ingen funksjoner er tilgjengelig uten pålogging, har vi holdt index-siden så enkel som mulig. Med dette har vi prøvd å gi målgruppen melding om at PP er gratis, og at pålogging er helt nødvendig.

Figur 2.3.7.1.1 Viser hvor et medlem kan registrere seg eller logge seg på.

Det er opprettet spørringer mot databasen som sjekker at et nytt medlem ikke tar

i bruk et brukernavn som allerede ligger inne. Hvis dette skjer vil brukeren få melding om dette på samme siden. Feilmeldinger er også laget hvis allerede eksisterende medlem taster feil brukernavn og/eller passord. Det samme gjelder hvis PP har stått i lang tid uten å ha blitt benyttet. Etter en bestemt tidsperiode så vil sesjonen hans bli slettet, og index-siden genererer en feilmelding.

Under følger utdrag fra koden som tas i bruk ved pålogging. Sidene som inneholder denne koden er såkalte skjulte sider, da de kun behandler data og videresender med en gang.

**Kode som sender lesepekeren til neste linje i koden ved feil**

```
<%on error resume next
```

**Inntastede verdier tas imot og legges i variable**

```
formBrN=request.form("brN")
```

```
formPassord=request.form("passord")
```

```
formBrN = replace (formBrN, "", " ")
```

```
formPassord = replace (formPassord, "", " ")
```

**Spørring som sjekker inntastet brukernavn og passord mot databasen**

```
s_sql = "Select mld From Medlem where brukernavn =" & formBrN & "" & _
" and passord =" & formPassord & ""
```

**Spørringen blir kjørt mot databasen**

```
set objRs = conntemp.execute(s_sql)
```

**Hvis spørringen finner kombinasjonen til medlemmet så opprettes sesjonsvariable som blir tatt vare på i en inkluderingsfil som heter session.asp.**

```
if not objRs.eof then
session("id")=objRs("mld")
session("medlem")=rstemp("brukernavn")
```

**Sender til dinside.asp hvor session.asp er inkludert.**

```
Response.redirect("../Nytt/dinside.asp")
```

**Hvis brukernavn og/eller passord er feil blir han stående i index. Parameteren feil blir sendt med. Denne genererer en feilmelding.**

```
else
Response.redirect("index.asp?feil=2")
end if
%>
```

Registrering av nytt medlem blir gjennomført på følgende måte:

**Inntastede verdier tas imot og legges i variable**

```
navn=request.form("fN")
etternavn=request.form("eN")
epost=request.form("eP")
brukernavn=request.form("brN")
passord=request.form("brP")
```

**Strenger med apostrof gjøres om slik at sql- spørringen takler den**

```
navn = replace (navn, "'", "''")
etternavn = replace (etternavn, "'", "''")
epost = replace (epost, "'", "''")
brukernavn = replace (brukernavn, "'", "''")
passord = replace (passord, "'", "''")
```

**Databastilkobling som for øvrig er inkludert i session.asp på alle medlemssider**

```
set conntemp = server.createObject ("adodb.connection")
katalog_filnavn = "DBQ=" & server.mappath("../Db/database.mdb")
conntemp.open "DRIVER={Microsoft Access Driver (*.mdb)}; "& katalog_filnavn
```

**Spørring som sjekker om ønsket brukernavn allerede finnes**

```
sql = "Select brukernavn From Medlem where brukernavn =" & brukernavn & ""
```

**Spørringen blir kjørt**

```
set objRs=conntemp.execute(sql)
```

Hvis brukernavn er opptatt blir han værende på index- siden. Parameteren feil gir feilmelding 1 som er "Du må velge et annet brukernavn. Det er i bruk!".

```
if not objRs.eof then
'id2=objRs("brukernavn")
Response.redirect("../Nytt/index.asp?feil=1")
objRs.close
set objRs=nothing
```

Hvis brukernavn er ledig så opprettes en medlemsprofil med alle inntastede verdier fra registrerings skjemaet.

```
else
sql="INSERT into Medlem (fornavn, brukernavn, passord, etternavn, ePost) values "
sql=sql & "("&navn&"", "&brukernavn&"", "&passord&"", "&etternavn&"", "
sql=sql & ""&epost&"" "
set rstemp=conntemp.execute(sql)
rstemp.close
set rstemp=nothing
```

Det nylig innlagte brukernavn og id blir hentet ut og lagt i sesjonsvariable. Disse verdiene blir tatt imot som forklart i forrige kodeeksempel. Deretter blir han videresendt til **dinside.asp**.

```
sql2="Select mld, brukernavn From Medlem where brukernavn =" &brukernavn& "" &_
" and passord =" &passord& ""
set rstemp=conntemp.execute(sql2)
session("id")=rstemp("mld")
session("medlem")=rstemp("brukernavn")
Response.redirect("../Nytt/dinside.asp")
```

### 2.3.7.2 SØKEFUNKSJONEN FOR SPILL, UTSTYR OG OMTALER

Som beskrevet i designdelen av dette dokumentet er venstre del i PP en bar med ulike valg som finnes for han. Her er søkefunksjoner for spill, utstyr og omtaler. I tillegg er det funksjoner for å hente skjemaer og egne annonser. Figuren under viser den venstre kolonnen (baren).



Figur 2.3.7.2.1 Venstre kolonne i PP. Her utføres store deler av interaksjonen.

Alle disse knappene kjører egne funksjoner. Søk etter spill og utstyr sendes til samme funksjon som viser resultatet avhengig av søkekriterier. Under kan du se koden bak spørrekriteriene for denne funksjonen.

VisResultat tar imot parametre fra et funksjonskall. Dette er enten for spill eller utstyr.

```
function visResultat(felt1,felt2,felt3,felt4,tabellnavn,kriterie,var1,var2,tekst1,tekst2)
```

funksjonen setter opp resultatet av søket i en tabell som inneholder en link til hele annonsen. Linken "mottaker" har destinasjon annonse.asp.

```
mottaker=" ../Salg/annonse.asp" sorter" settes inn i sql'en for å sortere søkeresultatet på
```

kriterie (som er enten spillnavn eller utstyrnavn) og på pris.

```
sorter=" order by "&kriterie&","pris "
```

Sql som opprettes med de medsendte parametrene fra funksjonskallet. Linjene er bygd opp med sql\_1, sql\_2 osv. for å få mindre og mer kompakt kode. Sql linjene kjøres da ettersom kriteriene i if- konstruksjonen under slår inn.

```
sql_1="Select " &felt1& "," &felt2& "," &felt3& "," &felt4& " from " &tabellnavn& "" sql_2=sql_1 & " where " &kriterie& " like '%" &var1& "%' "
```

```
sql_3=sql_2 & "and pris <=" &var2
```

```
sql_4=sql_1 & " where pris <=" &var2
```

En kobinasjon av if- tester som avhengig av medsendte parametre kjører ulike sql- linjer.

```
if sokeord = "Spill" and pris2 = 0 and hvilkenform = 3 then
```

```
set rtemp=conntemp.execute(sql_1 & sorter)
```

```
else if artikkel = "Utstyr" and pris = 0 and hvilkenform = 2 then
```

```
set rtemp=conntemp.execute(sql_1 & sorter)
```

```
else if sokeord <> "Spill" and pris2 = 0 and hvilkenform = 3 then
```

```
set rtemp=conntemp.execute(sql_2 & sorter)
```

```
else if artikkel <> "Utstyr" and pris = 0 and hvilkenform = 2 then
```

```
set rtemp=conntemp.execute(sql_2 & sorter)
```

```
else if sokeord <> "Spill" and pris2 <> 0 and hvilkenform = 3 then
```

```
set rtemp=conntemp.execute(sql_3 & sorter)
```

```
else if artikkel <> "Utstyr" and pris <> 0 and hvilkenform = 2 then
```

```
set rtemp=conntemp.execute(sql_3 & sorter)
```

```
else if sokeord = "Spill" and pris2 <> 0 and hvilkenform = 3 then
```

```
set rtemp=conntemp.execute(sql_4 & sorter)
```

```
else if artikkel = "Utstyr" and pris <> 0 and hvilkenform = 2 then
```

```
set rtemp=conntemp.execute(sql_4 & sorter)
```

```
end if
```

```
end if
```

```
end if
```

```
end if
```

```
end if
```

```
end if
```

```
end if
```

```
end if
```

Så lenge det finnes et resultat i databasen så skal funksjonen vise innholdet. Se fullstendig kodeeksempel i vedlegg B, kode.

```
if not rtemp.eof then
```

Her kommer listingen av resultatet





Når verdien av "salg" er "2" så vises feltene for salg av utstyr.

```
<%if salg = 2 then%>
<tr><td valign="top" align="left">
<p>Utstyr :<br></td>
<td valign="top" align="left">
<select name="navn">
<option></option>
<option value="PSOne"> Konsoll PS One </option>
<option value="PS2"> Konsoll PS2 </option>
<option value="joystick"> Joystick </option>
<option value="annet"> Annet </option></p>
</select>
</td></tr>
<%end if%>
<%if salg = 1 then%>
<tr><td valign="top" align="left">
<p>Konsoll :</td><td valign="top" align="left"><input type="radio" checked name="kon"
value="PS One"> PS One
<input type="radio" name="kon" value="PS2">PS2</p></td></tr>
```

Her hentes spillene fra databasen og legges i en rullegardinliste. Grunnen til dette er at vi ville styre hvilket spill som faktisk ble registrert.

```
<%'legger alle spill i en rullgardinliste
sql= "Select spillnavn from Spill order by spillnavn"
set rstemp=conntemp.execute(sql)%>
<tr><td valign="top" align="left"><p>Spillnavn :</td>
<td valign="top" align="left">
<select name="navn">
<option>&nbsp;</option>
<%do while not rstemp.eof%>
<option value="<%=rstemp("spillnavn")%>">
<%=rstemp("spillnavn")%>
</option>
<%går til neste post.Stenger det opprettede objektet
rstemp.movenext
loop
```

Her lukker vi objektet som har gjennomført spørringen mot databasen.

```
rstemp.close
set rstemp=nothing%>
</select>
</p></td></tr>
<%end if%>
```



Funksjonene som kan kalles på *dinsideRes.asp*. Disse kallene ligger i if- tester som styrer hva som vises til enhver tid. Som du ser av parametrene så utfører funksjonene oppgaver fra flere tabeller og kriterier.

Funksjonsnavnene har vi prøvd å gjøre selvforklarende og logiske.

```
call visTips("tips", "totID", "", "Tips")
call visTips("juksekode", "totID", "", "Kode")
call visTips("spillgjennomgang", "totID", "", "Walkthrough")
call visAnnonser("SpillSalg", "spillnavn", "pris", "Spillnavn", "Pris", "hentSS", "ssID", "Spill")
call visAnnonser("UtstyrSalg", "utstyryavn", "pris", "Utstyr", "Pris", "hentUS", "usID", "Utstyr")
call
visAnnonser("TipsOgTriks", "spillnavn", "totID", "Spillnavn", "Omtale", "hentTOT", "totID", "Tips")
call visAnnonser("SpillOmtale", "spillnavn", "omtale", "Spill", "Omtale", "bilde", "", "")
call visLinker(omtale, "SpillSalg", "ssID", "spillnavn", "Til salgs")%>
call visAnnonser("SpillOmtale", "spillnavn", "omtale", "Spill", "Omtale", "bilde", "soID", "")
call
visResultat("usID", "ePost", "utstyryavn", "pris", "UtstyrSalg", "utstyryavn", "artikkel,pris, "Utstyr", "Pris")
call visLinker(sokeord, "SpillOmtale", "soID", "spillnavn", "Spillomtale")
call
visResultat("ssID", "ePost", "spillnavn", "pris", "SpillSalg", "spillnavn", "sokeord,pris2, "Spill", "Pris")
```

### 2.3.7.5 INNLOGGING PÅ ADMINISTRATORSIDENE

Her vil administratoren møte en side som viser dato (inkluderer *adminDato.asp*), administratorlogo og en tofelts form (fig. 2.3.7.5.1) og en link til medlemssidene.

Figur 2.3.7.5. Loggininformen på administratorsidene.

Her skriver administratoren inn administratortnavn og passord, og dataene hun skrev inn blir sjekket mot databasen. Blir hun godkjent slippes hun inn til *adminHoved.asp*, blir hun ikke godkjent derimot blir hun sendt tilbake til *index.asp* hvor hun får beskjed at administratortnavn og/eller passord var feil.

Setter opp sql-setningen:

```
...
s_sql = "Select admId, admNavn From Administrator where admNavn ='" & formAdmin & "'" & " and
passord='" & formPassord & "'"
```

Utfører sql-setningen. Ved godkjenning kommer brukeren til *adminHoved.asp*:

```
set rtemp = connTemp.execute(s_sql)
if not rtemp.eof then
```

**Sesjon blir opprettet:**

```
session("administratorid")=rstemp("admId")
session("adminnavn")=rstemp("admNavn")
rstemp.close
set rstemp=nothing
response.redirect("adminHoved.asp")
```

*else*

**Blir hun ikke godkjent blir hun sent tilbake til index.asp:**

```
response.redirect("index.asp?kommando=ikkegodkjent")
```

...

**Visningen av sidene inneholder følgende test for å ta vare på sistnevnte tilfelle:**

...

```
else if kommando="ikkegodkjent" then
```

```
    response.write("<font color='cyan'>ADMINISTRATOR OG/ELLER PASSORD BLE IKKE  
GODKJENT!</font>")
```

...

### 2.3.7.6 OPPBYGNING AV ALLE ANDRE ADMINISTRATORSIDER

Alle administratorsider, bortsett fra *index.asp*, er bygget opp på følgende måte:

Først inkluderes filen *adminTopp.asp*.

Deretter kommer hovedinnholdet på siden

Til sist inkluderes filen *adminBunn.asp*.

*AdminTopp.asp* tar for seg følgende:

- Inkluderer filen *adminSession.asp*:  

```
<!--#include file="adminSession.asp"-->
```
  - Skriver deretter ut følgende:  

```
<html><head><title>Playstation Portal - Administrator</title>
```
  - Inkluderer deretter filen *adminDato.asp*:  

```
<!--#include file="adminDato.asp"-->
```
  - Linker så inn stilsettet som skal brukes:  

```
<link rel="stylesheet" type="text/css" href="../Inc/stil2.css">
```
  - Så legges *adminLogo.gif*:  

```

```
  - Skriver deretter ut en meny som rullegardinliste:  

```
<SCRIPT LANGUAGE="JavaScript">
  function goSelect(valget)
  {
    window.open(valget.options[valget.selectedIndex].value, "_top");
  }
</SCRIPT>
```
- ```

<form>
<SELECT onChange="goSelect(this)">
<OPTION VALUE="" SELECTED>Velg side...</OPTION>
<OPTION VALUE="adminHoved.asp">Hovedsiden</OPTION>
...

```

Tegner deretter opp en horisontal linje:

```
<hr>
```

Til slutt oppretter filen en kobling til databasen, som vil bli brukt på ALLE sidene:

```
<%
set conntemp = server.createObject ("adodb.connection")
katalog_filnavn = "DBQ=" & server.mappath("../Db/database.mdb")
conntemp.open "DRIVER={Microsoft Access Driver (*.mdb)}; "& katalog_filnavn
%>
```

adminSession.asp ser slik ut:

```
<%  
Sjekker at administratoren har en gyldig sesjon:  
adminnavn=session("adminnavn")  
administratorid=session("administratorid")
```

**Hvis sesjonen utløper sendes hun til logginnsiden:**

```
if administratorid = 0 AND adminnavn = "" then  
response.redirect "index.asp?kommando=utløpt"  
End if  
%>
```

adminDato.asp er programert i JavaScript. Vi har tatt utgangspunkt i kode vi har funnet på w3schools (se punkt 10.2 referanser: Internett), og modifisert denne.

Funksjonen bruker en array med navn for å ta var på ukedager og månedsnavn:

```
initArray("Søndag", "Mandag", "Tirsdag", "Onsdag", "Torsdag", "Fredag", "Lørdag", "Søndag");
```

Bruker Date() objektet for å fine (og lagre) datoer/dager:

```
var isnDays= new today=new Date();
```

Bruker også en finesse som korrigerer en feil visning i Netscape Navigator versjon 2 og 3 (se vedlegg A for mer informasjon)

adminBunn.asp sørger for å lukke alle objekter, samt lage en logg-ut-funksjon:

```
<%  
Lukker objekter  
rtemp.close  
set rtemp=nothing  
conntemp.close  
set conntemp=nothing  
%>  
Lager logg-ut-knapp:  
<center><form action="../Inc/adminBunnRes.asp" method=post>  
<input type="submit" value="Logg ut som administrator">  
</form></center>
```

adminBunnRes.asp tar imot fra foregående side, og sender brukeren tilbake til logg-inn-siden, index.asp, med en kommando som forteller index.asp at brukeren (administratoren) har logget ut:

```
<%  
session.abandon  
Response.redirect("../A/index.asp?kommando=loggut")  
%>
```

## 2.3.7.7 adminSalg.asp I DETALJ

Denne siden er en side som er meget oversiktlig på grunn av at den bruker funksjoner for henting/visning av informasjon. Siden ser ut slik det er illustrert i figur 2.3.7.7.1

Du er nå på siden for **spill-annonse** administrasjon.  
Trykk **utstyrSalg** for å gå til siden for salg av utstyr

**Følgende spill-annonser ligger i databasen:**

Velg **Slett** i tabellen for å slette spill-annonsen  
Velg **Rediger** i tabellen for å redigere spill-annonsen

Rediger	Slett	ssID	brukernavn	overskrift	spillnavn	beskrivelse	dato
Rediger	Slett	54	bredesen	Bra spill ja!	Ball Breakers	Har spilt det mye, men skiva er ikke slitt..	5/7/01
Rediger	Slett	23	Svennis	Dillespill !	Activision Classic	Dette er et spill man må være forsiktig med!	3/22/01
Rediger	Slett	61	bredesen	Kult spill	Ball Blazer Champions	Et helt nytt spill fra spillgigantene!	5/14/01
Rediger	Slett	59	bredesen	Kult spill	Area 51	Et helt gammelt spill!	5/14/01
Rediger	Slett	56	bredesen	Moro!	Ape Escape	Veldig morsomt spill om en ape som leker i hagen... (skikkelig gøy)	5/9/01
Rediger	Slett	81	plotzern	Salg av mitt beste spill!	Asteroids	Det er leit å selge et så kult spill, men når man ikke har til smør på brødiskiva da så....	dato
Rediger	Slett	43	Wergeland	Tøysespill fra Disney	A Bugs Life	Morsomt og med fin musikk!	3/29/01
Rediger	Slett	57	bredesen	Uspilt	Bubsy 3D	Spillet ligger fortsatt i pakningen!!	5/11/01
Rediger	Slett	58	bredesen	Wild!	Countdown Vampires	Et helt rått spill med mange skumle lyder og bra grafikk. Har vært redd mange ganger under dette spillet!	5/11/01

Figur 2.3.7.7.1. Administratorsiden for salg av spill (med link til utstyrSalg)

Siden inkluderer *adminFuncSalg.asp* i tillegg til de vanlige inkluderingene, og finner og viser informasjon ved å kalle funksjoner fra *adminFuncSalg.asp*:

**hvis spilltabellen skal vises:**

```
if kommando="spillSalg" then
    call visTabell("spill")
```

...

All kode som faktisk utfører arbeidet slik at brukeren ser det hun ser ligger da i *adminFuncSalg.asp*. Denne koden er MYE større enn dette ene funksjonskallet som kun sender med en parameter til funksjonen. Funksjonen er på til sammen 82 linjer! (se vedlegg A for fullstendig kildekode)

## 2.3.8 BEGRENSNINGER

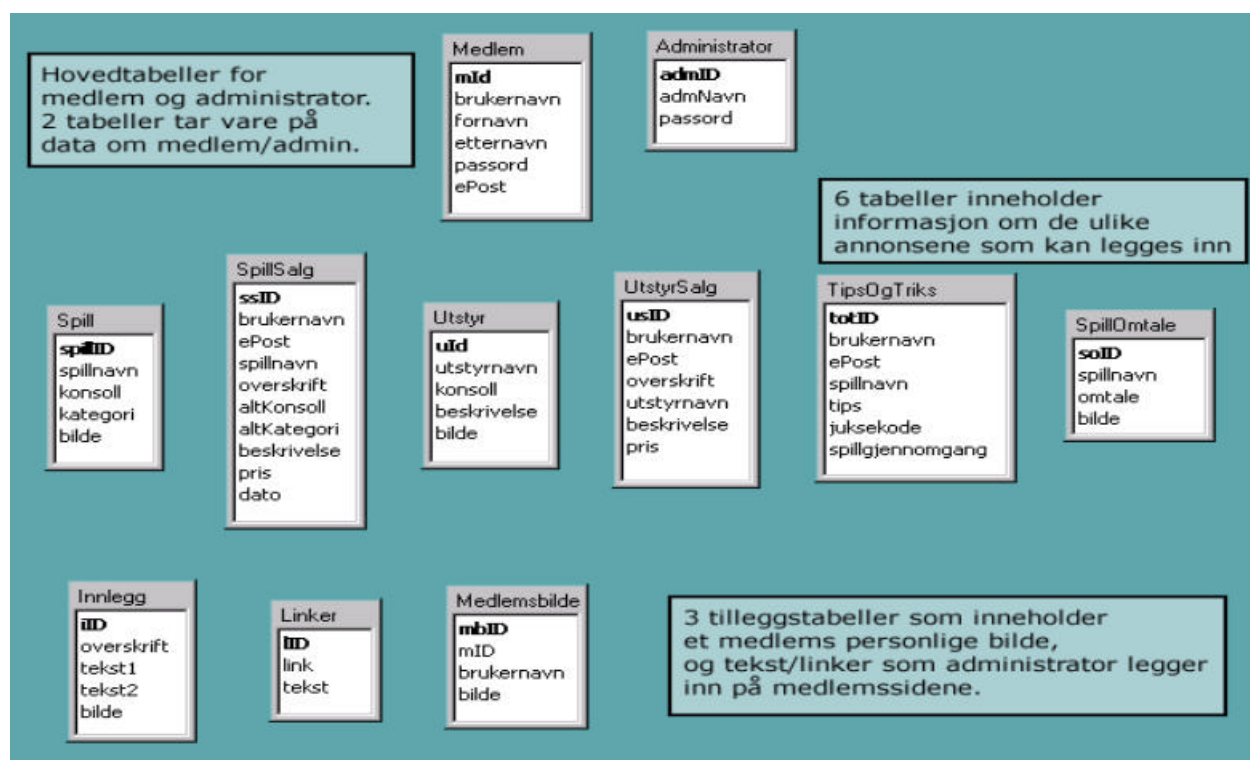
### 2.3.8.1 SOFTWARE ASPEKTER

Oppdragsgiver stilte ingen krav med hensyn til software standard og språk, derfor har vi valgt å bruke HTML 4.0, JavaScript 3.1 og ASP 2.0 m/VBScript 3.1 som implementeringsspråk. Til rapporter har vi laget egne standarder. Kombinasjonen av katalognavn og filnavn identifiserer filene i løsningen. Oppdragsgiver krevde ingen spesielle software pakker/verktøy, men anbefalte Microsoft InterDev.

### 2.3.8.2 DATABASE

Databasen som brukes i løsningen er en Access database som vi har kalt **database.mdb**. Den inneholdt færre tabeller ved første innlevering, og endringer er gjort i feltene i noen tabeller. Dette må ses i sammenheng med at vi har funnet måter å forbedre, samt at vi har opprettet nye tabeller fordi oppgaven ble utvidet ved vårt beslutningspunkt (se Gantt-skjema, vedlegg B).

For informasjon om tabeller i databasen, se figur 2.3.8.2.1.



Figur 2.3.8.2.1. Tabellene i databasen, samt feltene i disse (primærnøkker er uthevet).



**2.3.8.3 OPERATIVSYSTEM**

Webserver kjøres på Microsoft Windows NT. PWS er installert som testserver hjemme. Bruker er ikke avhengig av en bestemt type operativsystem, men det vil ikke bli gjort funksjonskontroller mot Macintosh eller Linux. Playstation Portal er laget på Windows NT 4.0 server med IIS 4.0 og Windows 98SE/ME med Microsoft Personal Web Server.

**2.3.8.4 TOLERANSER, MARGINER OG MULIGHETER/TILFELLER**

PP sin ytelse vil være avhengig av den enkelte server som systemet blir installert på. Dette er ikke beregnet i vårt prosjekt, men vi har tatt utgangspunkt i de servertypene som står oppført i punktet ovenfor(2.3.8.3) Å beregne kapasiteten til serveren som løsningen legges på, ligger derfor utenfor våre avgrensinger av prosjektet.

**2.3.8.5 DOKUMENTASJON**

Det blir ikke utviklet manualer eller opplæringssystemer for henne. Dokumentasjon er kun tilgjengelig fra prosjektrapporten. Dette mener vi er tilstrekkelig for å kunne bruke systemet da all informasjon som skal håndteres ligger i et forklarende webgrensesnitt.

**2.3.8.6 SYSTEM OG PROSJEKTTESTING**

Oppdragsgiver tar for seg avgjørende dokumenter og kode for å finne den mest optimale løsning. Dette skjer etter at gruppa har utviklet modulene. Tilbakemelding og diskusjon over telefon og e-post er stor del av testingen. I tillegg er det lagt ut demo- versjoner av ASP- sidene. Disse kan nås fra prosjektsiden: <http://higstud5.hig.no/playstation> (frem til 15. juni, 2001.)

**2.3.8.7 KRAV TIL SUPPORT, SERVICE OG VEDLIKEHOLD**

Service og vedlikehold er forbeholdt administrator. De som tar i bruk versjonen trenger ikke endre noe i selve oppbyggingen av strukturen, men står fritt til å videreutvikle eller utvide demoversjonen etter avtale med oss.

**2.3.8.8 KRAV TIL UTVIDELSER**

Da vårt produkt er en demoversjon står oppdragsgiver og eventuelle kunder ansvarlig for å utvide, hvis gruppa da ikke blir forespurt dette som ett frittstående prosjekt. Det er ikke satt restriksjoner for utvidelse av produktet. Utvidelse er derimot noe databasen ikke er optimalisert for.

**2.3.8.9 DEMOUTGIVELSE**

Det er ikke avtalt noen spesielle utgivelser underveis. Men prosjektet ligger på Internett slik at det var mulig for oppdragsgiver å se resultatene underveis. I tillegg til dette ble det lagt ut en demoversjon av prosjektet i uke 14. Dette ble gjort i forbindelse med innlevering av oppgave i faget klient- og serverside-programmering.

**2.3.9 AKSEPTANSEKRAV**

Playstation-Portal skal være en service til Playstation-brukere. Den skal gi medlemmer informasjon og kontaktmuligheter til andre som skal selge spill og utstyr. Alt som

legges ut på nett skal fungere så lenge det ikke oppstår feil som vi ikke råder over eller under noen omstendigheter kan ha kontroll på. Alle linker skal fungere. Informasjonen i databasen skal oppbevares i den fristen som hun setter.

### **2.3.10 PROSJEKTSTYRING, INKLUDERT KVALITETSSIKRING**

Forholdet mellom oppdragsgiver og oss har vært bra og vi har ofte hatt kontakt med dem. Kommunikasjonen har foregått stort sett elektronisk, men også noe over telefon. De første møtene var fysiske møter for å bli kjent, og for å finne felles arbeidsmål. Alle hovedmoduler og rapporter som ble produsert har blitt sendt til oppdragsgiver for godkjenning og utbedring. I tillegg hadde vi noen delte ansvarsområder som skulle være med på å øke kvalitetskontrollen. Det ble utarbeidet dokumentasjon på prosjektet fortløpende. Samarbeidspartner har vært en passiv partner, men som allikevel har gitt oss den nødvendige tilgangen til informasjon. Dette ved at vi fikk tilgang til en server hvor vi kunne finne nøyaktig og pålitelig informasjon om Playstation, spill, utstyr og artikler. (adresse til denne serveren finnes i punkt 10.2 referanse: Internett) Brukernavn og passord for aksess til denne serveren er ikke lenger gyldig da denne rapport levers.

### 3 ANALYSE OG DESIGN

Dette punktet viser i korte trekk hvordan vi lagde designet og arbeidsprosessen bak. Det endelige designet ble lagd fra og med uke 16, og derfor er all forklaring basert på de siste 5 ukene av prosjektet.

Oppdragsgiver ville at vi skulle lage en portal for spilleglade mennesker som trengte et sted å selge sitt brukte utstyr og sine spill. De vanlige funksjonene var satt opp av oppdragsgiver, mens en del av oppgaven var å legge til funksjonalitet som ville gjøre prosjektet mer helhetlig og omfattende. Programmeringsspråk på serveren ble fastsatt til ASP, men MS Access ble programmet vi brukte for å utforme databasen.

JavaScript er en scriptutvidelse av HTML som gir mulighet til å kontrollere HTML-siden og som gir respons på brukerhendelser uten å måtte kommunisere med en server. Dette scriptspråket ble vi også enige om å bruke for blant annet å sjekke inntastede verdier osv. Alt utseende med fargevalg, fonttype osv var det opp til oss å bestemme.

Nøkkelordet for utseende var portal. Oppdragsgiver ville at PP skulle ha samme oppbygning som andre, liknende portaler. Et eksempel vi fikk skissert var nettsiden [www.annonse.no](http://www.annonse.no). Med bakgrunn i dette startet vi med å finne sider på Internett som sto i stil med denne. Det vi endte opp med var en sideoppbygning som ligger på følgende URL: [www.w3schools.com](http://www.w3schools.com). (Se punkt 10.2 referanser: Internett) Denne siden er bygget opp av tre kolonner som er inndelt med informasjon i samme genre i samme kolonne. Dette er en oversiktlig oppbygning som også gjør at det er enkelt å inkludere hoveddeler som en egen kolonne.

Med dette som utgangspunkt tok vi for oss oppgavedefinisjonen og planla hvordan vår informasjon skulle innlemmes på siden. Dette var en prosess som kun gjaldt medlemssidene. For administratorsidene var ikke design et prioritert punkt, men oversikt, enkelhet og selvforklarende og raskt design kom i første rekke. Vi delte oppgavene på den måten at Kristoffer sto for sidene til administrator, mens Thorgrim lagde sidene for medlemmer. Allikevel holdt vi oss samlet på den måten at vi rådførte med hverandre i viktige punkter.

I designfasen ble strukturen satt opp, og fargevalg og skrifttyper valgt. Fargene varierte mye i starten da det var litt uenighet mellom oss og oppdragsgiver, men vi så fort at en kombinasjon skilte seg ut som den beste med hensyn til ergonomi. Det er to farger som brukes som basis. Hovedgrensesnittet er utviklet for at det skal være enkelt å bruke, oversiktlig og pent å se på.



Figur 3.1. Bildet viser oppbygningen av hovedsiden. De 3 kolonnene skal gi et ryddig preg.

I figur 3.1 kan du se hovedgrensesnittet for PP når han er logget på. Til venstre ser du en kolonne som blir en personlig menybar for å utføre de operasjoner som han ønsker. Her finnes tre søkefunksjoner som henter ut data fra databasen etter det søkekriteriet som skrives inn. I tillegg er det knapper som gir han mulighet til å hente ut sine egne annonser og utføre endringer på disse. Ved ønske om salg eller innlegg av tips så kan også dette gjøres fra den venstre kolonnen. Nederst er knappen som logger medlemmet ut av portalen. Under ser du en forklaring til det som forklares i dette kapittelet.



Figur 3.2. Strukturen på sidene i PP.

Ved å bruke den venstre delen så vil resultatet av interaksjonen vises i den grønne midtdelen av siden. Det er denne delen som til enhver tid forandrer seg. På hovedsiden finnes informasjon om antall annonser som medlemmet har i databasen, og det gis en forklaring til hvordan han kan laste opp sitt eget bilde som vil legge seg øverst i høyre del av siden. Dette for å gi medlemmet følelsen av å ha "sin egen side" i portalen. På figur 3.1 har han lastet opp et bilde av skyer på en blå himmel. Høyre del inneholder i tillegg linker til interessante sider som hun legger inn i modulen for administratorer.

Tekst ble satt opp på den måten at venstre kolonne fikk mindre skrift, både for å spare plass og for å gjøre det til en egen del på hovedsiden. Vi brukte Arial som skrifttype, størrelse 10, dette er inkludert skriften i rullegardinene. De andre delene fikk større skrift, og midtdelen følte vi ble mer oversiktlig og brukervennlig med størrelse 12. Dette viser at midten er det utskiftbare området som han skal bruke til utfylling og til å se på søkeresultater. Alle sidene er bygd opp med inkluderingsfiler som ligger i mappen **Inc**. Oppsettet av alle medlemssider ser derfor slik ut (se figur 3.2 som grafisk fremstilling av koden under):

**Session styrer tilgangen til sidene. Ved godkjent sesjon har et medlem tilgang. Session inneholder også oppkobling av databasen.**

```
<!--#include file="../../Inc/session.asp"-->
```

**Topp inneholder logoen med tilhørende javascript for nedtrekk på menyknappene. I tillegg ligger det browsertest og funksjon for dato.**

```
<!--#include file="../../Inc/topp.asp"-->
```

**Venstre del er den samme som i figur 2.3.7.2. Her ligger alle forms som styrer store deler av hans interaksjon med systemet.**

```
<!--#include file="../../Inc/venstre.asp"-->
```

Denne koden indikerer at innholdet på midten av siden kommer under denne innledningen.

```
<!--Her starter midtpartiet på siden ----->
<table border="1" width="100%" cellpadding="3" cellspacing="0" bgcolor="#336666">
<tr><td bgcolor="#336666">
<img src='../Inc/Bilder/banner.gif' width='490' height='70'>
```

Innholdet på hver enkelt side kommer her.....

Kontakt er den nederste delen av midtpartiet. Her ligger koden som viser e-post og telefonnummer til web- ansvarlig.

```
<!--#include file="../Inc/kontakt.asp"-->
<td valign="top" width="140">
```

Hoyre inneholder personlig bilde og linker til ulike sider som passer til PP. Delen har mye ledig plass og kan derfor lett utvides.

```
<!--#include file="../Inc/hoyre.asp"-->
```

Bunn inneholder ende- tag'er til html samt lukking av databaseforbindelsen.

```
<!--#include file="../Inc/bunn.asp"-->
```

For å se fullstendig kildekode til inkluderingsfilene, se vedlegg A

Fargevalgene ble som tidligere nevnt endret flere ganger i løpet av prosjektet, og vi endte til slutt opp med to websikre farger som vi mente sto bra til hverandre. Den midtre kolonnen av PP har fargekode **#336666**, mens alle andre steder har vi brukt fargekode **#003366**. Teksten valgte vi å gjøre hvit, da det ga god kontrast og teksten var lett leselig. Ved visning av data fra databasen kom vi fram til en tabell med to ulike gråtoner. Dette gjorde vi for å bryte litt med de to hovedfargene, og for at det var et ryddig oppsett av søkeresultatet.

Linker er gjort gule. Dette gjelder både for medlemssidene og for administratorsidene. I Internett Explorer bruker vi en hover-effekt som gjør at når musepekeren kommer over linken så blir den underlinjert og skarpere. Netscape er ikke kompatibelt med denne funksjonen, så NC-brukere mister denne effekten. I denne browseren blir linken rød når linken blir trykket (aktiv). Gule linker skilte seg ut på siden, og det passet fint til bakgrunnsfargene.

Påloggingssiden skiller seg litt fra IE til NC. I IE har vi lagt farge i bakgrunnen i tekstfeltene og på knappene. Dette fungerte ikke i NC, og på grunn av visning av tekstfelter i NC så mente vi at det ikke ble så fint å se på. Disse forskjellene er ikke kritiske, men går på små designmessige ønsker for PP. Oppdragsgiver og vi mener sidene tar seg best ut i Internet Explorer.

Administrator sidene bruker flere av de samme fargene som medlemssidene og samme stilsett. I tillegg til dette har alle sidene en overskrift i *color="lime"* av typen `<h2>`. All informasjon som ligger fast, er statisk og som ikke er til å forandre på er også skrevet med *color="lime"*. Variabel informasjon, som for eksempel data hentet ut fra databasen er skrevet med hvit skrift (dette er beskrevet i stilsettet). All informasjon som er direkte meldinger til brukeren (administratoren), blir skrevet med *color="cyan"*. Alle fargene som er valgt er såkalte websikre farger (browser-safe colors, se punkt 10.1.1, HTML for the world wide web, side 66). Det er som tidligere nevnt ikke lagt spesiell vekt på utseendet av administratorsidene. Disse skal i første omgang være, oversiktlige, raske og effektive. Det forventes en kort opplæring for bruk av systemet, og derfor er det ikke fullstendig forklaringer på alle sider til alle funksjoner på siden. Et eksempel på utseendet på administratorsidene kan sees i figur 3.2.

**Registrere ny administrator**

NY ADMINISTRATOR ER REGISTRERT

Gjeldende Administrator :

Gjeldende Passord :

Ny Administrator :

Nytt Passord :

Registrert!

**Følgende administratorer er registrert:**

test
birger
plotzingo
Leif

Figur 3.2. Eksempel på layout på administratorsidene.

## 4 KODING OG PRODUKSJON

I løsningen er det brukt ASP, JavaScript, CSS, SQL og HTML.

Grunnen til at vi har valgt å bruke ASP, er at det tilbyr opprettelse av dynamiske websider med omtrentlig samme funksjonalitet som CGI, PHP og Java Servlets. Likt disse er ikke ASP et eget, fullkomment programmeringsspråk, men en teknologi som ved hjelp av scriptspråk kjører kildekode på serveren og returnerer dette som HTML til klienten. Kildekoden skrives i en vanlig HTML-fil adskilt fra standard HTML-kode i blokker markert med `<% .....%>`. En fil som benytter ASP må bruke ekstensjonen ".asp" stedet for vanlig .html / .htm endelse for å fortelle serveren at den skal lete etter og kjøre all kode inne i en `<%...%>` blokk. Den vesentlige forskjellen mellom HTML og ASP er at:

- HTML leses og tolkes av browseren; hos klienten
- ASP leses og tolkes av en "ASP- motor" på serveren

En tradisjonell webside bygges opp hos klienten ved at denne først spør en server om en spesiell side ved å sende en request (forespørsel) til serveren inneholdende URLen til ønsket webside. Serveren leter frem aktuell fil angitt i URLen og returnerer denne som en strøm av HTML-kode. Når en klient spør serveren om en side med ekstensjonen ".asp" vil serveren lete etter ASP-script i fila. All slik kode vil prosesseres av ASP-motoren og returneres til serveren som standard HTML-kode. Denne koden returneres fra serveren til klienten som en helt vanlig strøm av HTML-kode. Dermed kan en hvilken som helst browser "lese" en ASP-fil. Generelt kan man si at ASP dynamisk lager HTML-sider. VB-script er benyttet for å skrive ASP-kode.

### 4.1 ORD OG UTTRYKK I ASP

ASP definerer 6 objekter. En kort forklaring på de mest brukte objektene (samt en funksjon) i løsningen følger:

#### 4.1.1 REQUEST

Er en forespørsel fra klient til server om visning av en webside.

**Request-objektet** i ASP bærer all tilleggsinformasjon fra klienten; det være seg fra en form (**request.form**), i en spørrestreng *request.querystring*, eller andre samlinger av nøkkel/verdi par som ASP definerer.

#### 4.1.2 RESPONSE

Er svaret serveren sender tilbake til klienten som resultat av en forespørsel.

**Response-objektet** i ASP tar seg av den informasjonen som skal sendes fra server til klient.

#### 4.1.3 RESPONSE.WRITE

Når informasjon i form av HTML kode sendes tilbake til klienten benyttes denne funksjonen (**method**). Denne funksjonen kalles med den informasjon som ønskes sendt som parameter. Eks. : *Response.Write("Hallo verden!")* legger "Hallo verden!" til strømmen av HTML fra serveren. En kortform av funksjonen er `<%= "Hei sensor!" %>` .



#### 4.1.4 SESSION

En **Session** kan defineres som det tidsrom en bruker aktivt benytter websidene tilhørende en applikasjon. Slik vil det kunne lagres informasjon i et **Session-objekt** om hver enkelt brukers samtidige interaksjon med applikasjonen.

#### 4.1.5 SERVER

**Server-objektet** tilbyr en rekke funksjoner for å kunne bygge opp dynamiske sider med ASP, og har ingen direkte effekt på visningen av web-sidene.

Eksempler:

Opprette ActiveX komponenter med [Server.CreateObject\(\)](#).

Finne fysisk path til en virtuell katalog på server med [Server.MapPath\(\)](#)

#### 4.1.6 JAVASCRIPT

Grunnen til at vi valgte å bruke JavaScript var at det ga mulighet til å bygge dynamiske websider. I tillegg inneholder det en del innebygde funksjoner og objekter som kunne benyttes til vårt formål. JavaScript-koden ligger i et HTML-dokument og kjøres i browseren til brukeren. Dette er blant annet med på å gi bedre, mer brukervennlig utfylling av ulike skjemaer i PP.

#### 4.1.7 HTML

Løsningen er bygd opp med HTML som basis. Denne teknologien gis ikke mer forklaring her.

#### 4.1.8 SQL

Er et standardisert språk som brukes til å kommunisere med en database. Vi har benyttet dette for å hente fra og skrive til databasen. Innebygde spørringer er blitt brukt der disse henter informasjon fra flere tabeller, såkalte krysspørringer. Spørringer der det bare blir hentet informasjon fra en tabell er en tekststreng som inneholder SQL-spørringen. Dette blir brukt i ASP-koden.

## 5 TESTING OG KVALITETSSIKRING

Testing har blitt utført kontinuerlig under koding. Vi har som oftest testet selv, men har også fått hjelp av andre, blant annet medstudenter og familie. Oppdragsgiver har foretatt tester på betaversjon og endelig innlevering. Vi har brukt ulike nivåer innenfor testing:

- Jevnlig og hurtig sjekking under koding
- Testing av hele moduler (som søkefunksjonens samtlige kriterier)
- Utenforstående prøver det vi lager for å se reaksjoner og uforutsette hendelser
- Full test av demoversjonen
- Komplettest ved innlevering av produktet

PP har vært testet i to browsere, Internett Explorer 5 og Netscape Communicator 4.7. Disse er installert på skolen og har vært en pekepinn på om designet ble likt for begge typer browsere. Det har vært et viktig punkt for oss at sidene skal se så like ut som mulig i begge typer browsere. Noe av ulikhetene skyldes for det meste definerte størrelser som ikke stemmer overens i de to browserne. Tester i andre nettlesere som for eksempel Opera er ikke blitt gjort, da markedsandelen er desidert størst fordelt på IE og NC.

Kvalitetssikring er gjennomført i form av møter, samtaler, eksperthjelp (les: veileder og oppdragsgiver) og statusrapporter. Møtene vi har hatt med veileder har vist oss hvor nivået må ligge for å få et godt resultat. I tillegg utarbeidet vi en enkel sjekklister for ASP og HTML-kode som ble brukt i avslutningen av prosjektet. Kommentering av kode har også vært et punkt som vi mener er med på å kvalitetssikre og simplifisere programmeringen som ligger bak.

Det er kontinuerlig foretatt backup av samtlige filer i prosjektmappen. Vi har også lagt prosjektet på ulike servere i egne dato-mapper som viser tidspunktet for kopieringen. Noe kode er også sendt til oppdragsgiver, som har kommet med forslag for å optimalisere den til hurtig og enkel eksekveringsdyktig kode.

## 6 REALISERING / INSTALLASJON

Prosjektet har en vedlagt CD-ROM som inneholder PP og all dokumentasjon bak det endelige resultatet. Vi har ikke tatt med programvare som browsere, PWS eller andre nødvendige programmer. Dette forutsetter vi at målgruppene har installert fra før av, samt at nødvendig programvare kan lastes ned gratis over Internett. Katalogen med PP legges i rota av den serveren som skal brukes. All linking mellom filer vil da være i orden grunnet bare relative linker innad i prosjektet og PP er derfor klar til bruk. Vi gjør oppmerksom på at e-post generatoren som er en modul som administratorer bruker, er et program som ligger installert på en server her ved HiG. Etter endt prosjekt vil ikke denne adressen lenger være tilgjengelig, men programmet er gratis og kan installeres et annet sted. Løsningen er per i dag ikke ferdig implementert. Løsningen er tenkt implementert for å vise at å sende e-post til medlemmer er enkelt og hendig for en administrator. Under følger en enkel forklaring for hvordan ta i bruk PP for henne:

Pålogging gjøres fra følgende URL:

/A/index.asp (pr i dag: <http://higstud5.hig.no/playstation/a/index.asp>)

Det vil med systemet følge med ett brukernavn med ett tilhørende passord som kan brukes til å komme inn som administrator. Etter dette er gjort så kan vedkommende opprette tilstrekkelige brukernavn og passord ut fra eget ønske. Andre modifikasjoner blir ikke spesifisert da det anses som uviktig. En administrator med tilstrekkelig data-kunnskap kan selvfølgelig gå rett inn i databasen og gjøre endringer, men dette forklares ikke her. For demonstrasjon kan brukernavnet: **test** brukes som administratornavn og **testen** brukes som passord for å teste løsningen. Denne logginn- løsningen vil selvfølgelig fjernes hvis systemet skal legges ut på nett for en eventuell kunde.

## 7 KONKLUSJON OG DRØFTINGER

Når vi nå er ferdig med oppgaven kan vi se tilbake på en lang prosess med mange veivalg og mange vanskelige avgjørelser. De fleste oppgaver har vi løst mer eller mindre greit og vi syntes vi hadde god oversikt over oppgaven og den løsningen fram til slutten av april. Da fikk vi av veileder vite at vi hadde altfor mye kode og for lite funksjonalitet. Da sto vi ved vårt vanskeligste valg gjennom hele prosessen: Skulle vi legge til mer funksjonalitet til det gamle systemet, eller skulle vi gjøre en snuoperasjon og gjøre om alle sidene til en mer profesjonell løsning fra bunnen av. Etter en konfrontasjon med oppdragsgiver valgte vi det siste, og satset fullt og helt på dette. Det skulle vise seg å bli store mengder ekstra arbeid, og vi hadde under en måned å realisere dette på. Det kan synes hasardiøst å satse på en så stor omveltning når det var så liten tid igjen, men vi ville ikke levere fra oss et resultat vi i realiteten ikke var fornøyd med. Vi ville greie det, og vi ble så absolutt fornøyd med resultatet kontra de "gamle" sidene. I tillegg til å optimalisere mange sider med kode fikk vi også lagt til noen nye funksjoner. Ikke alle de NYE funksjonene ble 100% ferdig til innlevering, men vi ser oss likevel meget godt fornøyd med resultatet siden disse funksjonene ikke var planlagt fra begynnelsen og derfor kan sees på som en slag bonus. Siden et av målene med prosjektet egentlig i bunn og grunn var å lære mest mulig ASP er vi i dag meget glade med valget vi tok. Vi har lært å bruke ASP på en mer programmeringsmessig effektiv måte, samt lært hvordan sider bør bygges opp fra grunnen av. Med løsningen vi har valgt nå er det meget lett for en som kan litt ASP å lage nye sider, og inkludere disse til portalen.

### 7.1 MEDLEMSSIDENE

Det er her funksjonaliteten og designet har innvirkning på medlemmene. For disse sidene var det definert noen standardoperasjoner som han skulle kunne utføre. Oppgaven hadde mange muligheter, men i og med at vi var kun to personer så måtte vi sette en del grenser. Det må presiseres at dette er kun på grunn av tidspress. For at en portal som denne skulle bli interessant så måtte mulighetene være gode for et medlem. Valgene trengte å være lett tilgjengelige, med lettfattelig og forståelig utseende og oppsett. Sidene ble tenkt bygd opp på den måten at et medlem registrerte seg med en e-post som da ble lagt med i alle annonser som han la inn. Når da andre søkte etter disse annonsene så ville e-post adressen være den eneste muligheten for kommunikasjon. Etter at vi gjennomførte en snuoperasjon i uke 16 så har alle medlemssidene fått nytt design og en annen ny og bedre oppbygning av koden. Det er siden etter snuoperasjonen som blir beskrevet her.

For medlemssidene ble følgende deler satt opp som funksjonalitet:

- 1 Registrere seg med personlige opplysninger som blant annet navn, brukernavn og e-post adresse
- 2 Hovedside med generell informasjon. Her er det mulighet for å legge inn et personlig, valgfritt bilde. Siden inneholder også informasjon om medlemmets annonser i databasen. Sideoppsettet er ellers likt på alle sider, med tre kolonner som deler opp siden i naturlige bolker
- 3 Legge inn spill, utstyr og tips
- 4 Endre og slette spill, utstyr og tips
- 5 Søke etter spill, utstyr og omtaler
- 6 Visning av hele annonser på spill, utstyr og tips
- 7 Se nyheter. De 5 siste innlagte spill- og utstyrsannonsene listes opp
- 8 Kommunisering med andre medlemmer ved hjelp av e-post linker
- 9 Få oversikt over aktuelle linker som passer til en portal- side
- 10 Ved søk på spill vises aktuelle linker til omtaler av disse spillene
- 11 Ved søk på omtaler vises aktuelle linker til spill- salg

Disse funksjonene var fra starten av mindre og enklere, med fokus på det viktigste, nemlig å kunne legge ut spill og utstyr for salg. Punktene over er gjennomført, men vi har også satt opp punkter som hadde vært nødvendige hvis prosjektet hadde vært større.

- 12 Et medlem kan endre sine egne personlige opplysninger som brukernavn og passord. Han skal ha en profil som er enkel å modifisere
- 13 Søk etter tips og walkthroughs. Dette er et punkt som må være med, men som er utelatt på grunn av tidspress. Da dette er veldig likt andre søkefunksjoner som allerede finnes så er dette utelatt

Realiseringen av punktene ble basert på den første perioden av prosjektet som strakk seg frem til påsken i uke 15. Etter dette kom vi i en situasjon hvor koden måtte endres så mye at det var lettere å starte på nytt med ryddig, strukturert oppsett av sidene. Dette ble gjennomført ved å lage inkluderingsfiler av alt som gikk igjen på alle sider. Slik ble koden mindre og mer oversiktlig. I tillegg ble alle medlems- interaksjoner lagt i funksjoner som blir kalt når han utfører en brukerhandling. De fleste funksjonene er lagt i en fil som heter *funksjoner.asp* Der er hver funksjon kommentert slik at det er lett for oss å finne fram i koden. Andre filer som brukes ligger i katalogene *Nytt* og *Salg*.

Det er også lagt til sider som gjør at han kan legge inn sitt personlige bilde som vil vises på samtlige sider. Denne koden er hentet fra en zip-fil på URL:  
[http://www.hig.no/at/data/client\\_server/asp\\_linker.phtml](http://www.hig.no/at/data/client_server/asp_linker.phtml) (20.04.01). (Se punkt 10.1 Referanser: Litteratur) Koden er modifisert og optimalisert for PP.

## 7.2 ADMINISTRATORSIDENE

Dette er en samling sider som skulle brukes for å oppdatere og vedlikeholde viktige deler av Medlemssidene. Det ideelle tilfellet ville være hvis "alt" på medlemssidene kunne vedlikeholdes, oppdateres, sensureres og modifiseres. Dette er et tenkt tilfelle som det selvfølgelig er meget vanskelig å implementere. Vi tok utgangspunkt i hvilke

funksjoner som var vanlig på lignende sider, og hvilken funksjonalitet vi fant fornuftig å inkludere. Denne listen med funksjoner har vokst etter hvert som medlemssidene har blitt ferdig, og består ved innlevering (23.05.01) av i alt elleve forskjellige sider. I tillegg til disse har vi seks ASP-sider som ligger i Inc-katalogen.

Målet var å kunne behandle følgende tilfeller:

- 14 En innloggingsside hvor administrator måtte bli godkjent via databasen for å få tilgang til administratorsidene. Denne skulle også opprette en sesjon
- 15 Hovedside med aktuell informasjon hentet fra databasen
- 16 Legge til nye spill til databasen
- 17 Slette spill fra databasen
- 18 Lage nye, redigere og slette spillomtaler av spill
- 19 Vise oversikt over medlemmer, samt mulighet for å slette disse enkeltvis
- 20 Se oversikt over spill og utstyr som er til salgs, og mulighet for å slette annonser
- 21 Oversikt over tips og triks, samt mulighet for å redigere og slette disse
- 22 Registrere en ny administrator med ekstra sikkerhets loggin
- 23 Slette en administrator
- 24 Legge til nye og slette gamle linker
- 25 Sende én e-post til alle medlemmene automatisk
- 26 Endre dagens tekst

Av disse tilfellene hr vi klart å realisere punkt 14 til 24. Punkt 25 og 26 har vi ikke rukket å implementere på grunn av sykdom i gruppa. Disse punktene ble derfor nedprioritert til fordel for å gjøre allerede ferdige punkter bedre kodemessig, og mer ryddige. Det må også poengteres at disse sidene ikke er optimale i forhold til slik de kunne være, men at heller ikke alle sidene var planlagt som en del av løsningen ra starten av, men har blitt lagt til for å imøtekomme veileders forslag til mer funksjonalitet.

For videre arbeid med sidene vil det være naturlig å legge til funksjonalitet slik at administrator kan velge farger på linker, bakgrunn osv, samt endre på annen layout. Hadde vi hatt bedre tid kunne disse elementene vært implementert, men vi har valgt å konsentrere oss om de funksjoner vi har realisert. Noen av sidene kunne helt sikkert hvert annerledes i forhold til oppsett, design og layout, men som omtalt i punkt 3. er dette ikke prioritert på administratorsidene. Flere av sidene har også forholdsvis lik funksjonalitet og kunne med fordel blitt delt opp i funksjonsfiler som kunne legges ved sidene. Dette for å gjøre koden mer oversiktlig, minske total kodemengde, samt få en bedre programmeringsmessig løsning på det hele. Dette er gjort på siden *adminSalg.asp* hvor siden inkluderer en annen side, *adminFuncSalg.asp* som inneholder alle funksjoner som siden trenger. Disse mottar attributter som forteller funksjonen hva den skal gjøre, og med hvilke data. Dette burde være standarden for alle sidene som brukte elementer som var felles med andre sider, men av tidsmessige årsaker har ikke dette blitt prioritert på administratorsidene. På medlemssidene har dette blitt gjort.

## 8 KRITIKK OG EVALUERING

## 8.1 KRITIKK AV OPPGAVEN

Oppgaven vi fikk fra oppdragsgiver var ganske godt definert, men vi fikk likevel forholdsvis frie tøyler til hvordan vi ville løse oppgaven. Vi skulle lage sider for Playstation-spillende surfere, men også sider for å vedlikeholde og overvåke disse. Implementeringsmessig kunne vi gjøre som vi ville, og dette gjaldt også design og implementering av egne forslag til funksjoner. Vi har gjennom hele prosessen holdt løpende kontakt med oppdragsgiver, som igjen har kommet med forslag til endringer, men også ris og ros.

Rammen rundt oppgaven var gitt, men detaljene var ikke definert. Dette har gitt oss stor handlefrihet, men også ført til at oppgaven til tider har vært svært utfordrende å arbeide med. Vi har i stor grad måttet finne ut av hvilke krav som skulle spesifiseres, for så å spesifisere dem. Vi har hatt få konkrete forslag til innhold i de enkelte moduler og har måttet fremskaffe så godt som all informasjon til disse modulene selv.

Planleggingsfasen var i utgangspunktet god, med en periode hvor vi definerte de første retningslinjene for oppgaven. Ved å arbeide med noe man ikke innehar kunnskap om ved starten av et prosjekt må man regne med at planleggingen i forkant kanskje ikke helt holder mål. Dette er nok et punkt som blir lettere i en arbeidssituasjon, hvor du vet hva du kan og hvor lang tid du sannsynlig vis vil bruke.

I starten ble funksjonalitet foretrukket framfor ryddighet og rask kode. Dette førte til en periode hvor det ble mye klipp og lim og at samme kode i realiteten gikk igjen på mange steder. I forbindelse med faget klient- og serversideprogrammering kom vi inn i en modningsprosess hvor vi sakte men sikkert skjønte mer av den riktige og beste måten å bygge opp en ASP-løsning på. Vi tar selvkritikk på mange områder, og innrømmer at en slik oppgave er en prosess som gir mye kunnskap og erfaring fra startfase til slutfase. Vi har lært at strukturering av både oppgave og løsning er alfa omega i et prosjekt med en viss størrelse, og at det er viktig å planlegge godt. Deler vi hadde planlagt godt og som var godt gjennomtenkt gikk mye lettere å implementere, samt at vi støtte på langt færre problemer med disse løsningene.

Når det gjelder videre drøfting av det endelige produktet, se også punkt 7.

## 8.1 EVALUERING AV GRUPPA

Gruppen har bestått av to studenter ved studiet Høgskolekandidat i Data- og Multimedieteknikk ved Høgskolen i Gjøvik. Vi har forskjellig bakgrunn innen datafaget, og dette har vært nyttig for oss gjennom arbeidet med prosjektet. Tross forskjeller innenfor kompetanse har vi begge vært stort sett vært enige om hvilke oppgaver som skulle prioriteres til en hver tid, samt hvem som skulle gjøre hvilke oppgaver.

Begge to hadde interesse av å lære mer om oppbyggingen av websider, kommunikasjon med databaser, og aspekter og muligheter rundt dette. I etterkant av et prosjekt vi hadde høsten 2000, fikk vi tilbud om hovedprosjektoppgave fra det samme firmaet som hadde vært oppdragsgiver for oss da, nemlig DMT-Internett AS. Denne oppgaven dreide seg om de temaene vi ønsket å fordype oss i. Vi takket derfor tidlig ja til prosjektet, men ettersom firmaet fikk nye eiere og medarbeidere, ble det ikke aktuelt å ha prosjekt med DMT som oppdragsgiver lenger. To av medarbeiderne dannet imidlertid et nytt selskap, "Web Annonse Norge", og dette ble vår oppdragsgiver ettersom det var deres ide om prosjektet i utgangspunktet.



## 9 ORGANISERING

### 9.1 ORGANISERING OG FORDELING AV ARBEID

Siden prosjektgruppa besto av kun to personer, fant vi i begynnelsen av prosessen ut at vi ikke ville ha én leder for prosjektet, med heller ha et likeverdig ansvar for det endelige resultat. Vi har byttet på å være møteledere for møtene vi har hatt, samt at begge har vært med på å skrive referater fra møter, avtale møter med sensor og gjøre alt forefallende administrative oppgaver.

Selv om vi hele tiden har hatt et delt ansvar for alle sidene ved prosjektet, valgte vi likevel å dele det hele i to. Thorgrim har hatt hovedansvaret for Medlemssidene, men Kristoffer står som hovedansvarlig for Administratorsidene. Tross dette skillet vil vi poengtere at ansvaret for hele oppgaven har ligget hos begge.

### 9.2 PROSJEKT SOM ARBEIDSFORM

Gjennom prosessen med prosjektarbeid har vi fått erfare både hvor effektivt dette kan være som arbeidsform, men også at det avhenger av visse faktorer. Det er et ordtak som sier at "Godt forarbeid er halve jobben", og dette passer godt inn under prosjekt som arbeidsform. De delene vi hadde planlagt godt gikk meget raskt å implementere, mens vi støtte på en del uforutsette problemer ved deler vi ikke hadde planlagt like godt. I og med at vi hadde en stor snuoperasjon i slutfasen av prosjektet, sier det seg selv at ikke alle aspektene ved denne ble planlagt like godt som vi kunne ønske; Det var rett og slett ikke tid nok.

Vi har vært to personer på dette prosjektet, og det har både sine fordeler og ulemper. En av de desidert største fordelene er at det har vært veldig lett å opprettholde en god kommunikasjon hele veien gjennom prosjektet. Det var få mennesker som skulle ha et ord med i laget, og de få uoverensstemmelsene som har vært, har latt seg løse enkelt og raskt. Ulempen med å være kun to personer på et såpass stort prosjekt er hvis det skjer noe uforutsett angående begrensede ressurser, som for eksempel tid. Da Kristoffer var syk 2 ganger, i til sammen nesten 2 uker, ble det mye mer arbeid for Thorgrim enn det som var planlagt. Heldigvis lot dette seg ordne, selv om vi kom et par uker etter oppsatt tidsplan. Hvis en av oss hadde blitt fraværende over en lengre periode, hadde vi med stor sannsynlighet fått store problemer med å levere det ønskede resultatet innen tidsfristen. Hadde vi derimot vært flere medlemmer i prosjektgruppa hadde det blitt mindre å ta igjen på hver og en, og den resterende arbeidsmengden kunne blitt fordelt på flere enn bare et menneske.

## 10 REFERANSER

### 10.1 Litteratur

#### 10.1.1 HTML 4, for the World Wide Web, Fourth Edition:

Visual Quickstart Guide

Elizabeth Castro

Utgivelsesår: 2000

ISBN: 0-201-35493-4

#### 10.1.2 Aktive Server Sider

Et kompendium av Øivind Kolloen

Utgivelsesår: 2001

ISBN: Lokalt utgitt ved HiG. Intet ISBN.

### 10.2 Internett

<http://www.learnasp.com/>

Sist besøkt: 21.05.01

<http://www.w3schools.com/>

Sist besøkt: 21.05.01

[http://www.hig.no/at/data/client\\_server/index.phtml](http://www.hig.no/at/data/client_server/index.phtml)

Sist besøkt: 21.05.01

<http://www.gjensidige.no/elcon>

Sist besøkt: 21.05.01

Vi har også funnet informasjon på undersider av alle disse sidene.

### 10.3 VEDLEGG

Vedlegg A	-	Fullstendig kildekode til eksempelsider
Vedlegg B	-	Gantt-skjema_og_framdriftsplan
Vedlegg C	-	Statusrapporter
Vedlegg D	-	Møtereferater (finnes ikke på CD-ROM da de er skrevet for hånd)
Vedlegg E	-	Loggbok
Vedlegg F	-	Timeliste
Vedlegg G		CD-ROM