

Hovedprosjekt:

**Gratulasjonshilsninger for
A-Pressen Interaktiv**

Forfattere:

Tom-André Skar
Atle Gulbrandsen
Espen B. Mikalsen

Dato:

19.mai 2005

Sammendrag

Tittel:	Gratulasjonshilsninger for A-Pressen Interaktiv.
Nr/Dato:	1/19.mai 2005.
Forfatter(e):	Tom-André Skar, Atle Gulbrandsen, Espen B. Mikalsen.
Veileder:	Rune Lossius.
Oppdragsgiver:	A-Pressen Interaktiv.
Kontaktperson	Torild Akre, Oppland Arbeiderblad.
Nøkkelord:	Gratulasjonshilsning, nettavis, ASP.NET, C#.
Antall sider:	115
Antall vedlegg:	5
Tilgjengelighet:	Åpen
Sammendrag:	<p>Nettansvarlig hos OA, Torild Akre, var tidlig ute med ideen om en gratulasjonstjeneste implementert i avisens nettutgave, der brukerne kunne legge inn bilder og hilsninger til sine bekjente. Ideen ble satt ut i live gjennom et hovedprosjekt ved HiG, våren 2003. Prosjektets tittel var "A-Pressen Interaktiv – På nett med ungdom" og hadde som mål å utvikle flere forskjellige tjenester rettet mot ungdommen. På grunn av at A-Pressen i løpet av våren og sommeren 2003 skiftet til den JAVA-baserte teknologien, Escenic, ble det ferdige produktet aldri klart til implementering. På samme tid kom også VG med sin versjon, noe som også gjorde det lite motiverende å jobbe videre med applikasjonen.</p> <p>Siden forrige løsning aldri ble ferdigstilt vil OA nå på nytt forsøke å få implementert gratulasjonsløsninger i nettavisen. Denne gangen skal det i tillegg fokuseres på at annonsen automatisk skal sendes inn til papirutgaven hvis dette er ønskelig fra brukeren. Betaling av annonse til både nett- og papirutgave skal kunne gjøres over nettet.</p>

Forord

Hovedprosjektet Gratulasjonshilsninger for Oppland Arbeiderblad og A-Pressen Interaktiv er et avsluttende prosjekt ved den treårige ingeniørutdanningen på Høgskolen i Gjøvik våren 2005.

Prosjektets deltakere har vært Tom-André Skar, Espen Bårdsnes Mikalsen og Atle Gulbrandsen.

Målet med prosjektet er å lage en gratulasjonsapplikasjon der nettavisens brukere skal kunne legge ut gratulasjoner og andre typer hilsninger på nettet og i avisens papirutgave. All administrasjon av hilsninger skjer gjennom vårt system, mens betalingen skjer gjennom det eksterne systemet Payex.

Ideen ble presentert for Høgskolen i Gjøvik ved Torild Akre som nå er vår kontaktperson i Oppland Arbeiderblad. Oppdragsgiver er A-Pressen Interaktiv.

Underveis i prosjektet har vi fått god hjelp og veiledning fra flere personer som vi gjerne vil rette en stor takk til her. Vi vil takke Nina Kristine Utne hos Payex og Webhuset.no sin support som har hjulpet oss med testing og svar på spørsmål. Vi må få takke veileder Rune Lossius for veiledning og gode innspill, vår gode venn Erik Wien for hjelp når vi har stått fast i C#, Mari Graftsrønningen for hjelp med design og Torlid Akre for positive og motiverende tilbakemeldinger.

Gjøvik 19.mai 2005.

Tom-André Skar

Espen B. Mikalsen

Atle Gulbrandsen

Innhold

1	Innledning.....	11
1.1	Bakgrunn.....	11
1.2	Målgruppe for rapporten.....	11
1.3	Formål med prosjektet	11
1.3.1	Hovedmål.....	11
1.3.2	Prosessmål.....	12
1.3.3	Effektmål	12
1.4	Gruppas arbeidsformer	12
1.5	Egen bakgrunn og kompetanse	13
1.6	Dokumentasjon, konfigurasjon og versjonsstyring	13
1.6.1	Dokumentasjon	13
1.6.2	Konfigurasjon og versjonsstyring	13
1.7	Teknologi.....	13
1.8	Øvrige roller	14
1.9	Organisering av rapporten	14
2	Kravspesifikasjon	17
2.1	Introduksjon	17
2.1.1	Bakgrunn	17
2.1.2	Kort om krav til systemet	17
2.1.3	Avgrensinger.....	18
2.2	Usecase.....	18
2.2.1	Usecase diagram.....	18
2.2.2	Usecase beskrivelser.....	19
2.3	Supplementær kravspesifikasjon	22
2.3.1	Ytelse.....	22
2.3.2	Sikkerhet.....	22
2.3.3	Operasjon av systemet.....	22
2.3.4	Logging og feilrapportering	22
2.3.5	Utvidelser.....	23
3	Analyse og Design	25
3.1	Introduksjon	25
3.1.1	Generelle mål.....	25
3.2	Funksjonelt design.....	25
3.3	Databasedesign	26
3.4	Datastruktur.....	27
3.5	Funksjonell struktur - brukerdel.....	28
3.5.1	Navigasjonskart.....	28
3.5.2	Sidene	29
3.6	Funksjonell struktur - administrator del	36
3.6.1	Navigasjonskart.....	36

3.6.2 Sidene	37
4 Implementering, koding og produksjon.....	43
4.1 Utviklingsmiljø	43
4.2 Programvare.....	43
4.3 Hardware	43
4.4 Valg av verktøy.....	44
4.4.1 Programmeringsverktøy	44
4.4.2 Kodestandard	44
4.4.3 Kommunikasjon mot database	45
4.4.4 Kode eksempler fra Database.cs	45
4.5 Cookies	46
4.5.1 Kode eksempel.....	46
4.6 Hjelp.....	47
4.7 Bildehåndtering	47
4.8 Payex	49
4.8.1 Generelt	49
4.8.2 Payex i vårt system	50
4.8.3 Implementering.....	50
4.8.4 Kommunikasjonen mot Payex	52
4.8.5 SSL	52
4.8.6 Kodeeksempler	53
4.9 Saxotech	54
4.9.1 Hva er Saxotech?	54
4.9.2 Implementering.....	54
4.9.3 EXIF.....	54
4.9.4 Kodeeksempler	55
4.10 Escenic.....	56
5 Testing og kvalitetssikring.....	57
5.1 Kompatibilitet	57
5.2 Kvalitetssikring	57
5.2.1 Sikring av data og backup	57
5.3 Testing.....	58
5.3.1 Testing av kildekode.....	58
5.3.2 Testing av bruker- og administrasjonsdel	58
5.3.3 Testing mot Payex	58
5.3.4 Testing mot Saxotech	58
6 Installasjon og realisering.....	59
6.1 Introduksjon	59
6.2 Installasjon.....	59
6.3 Vedlikehold	60
7 Beskrivelse av utviklingsprosessen.....	61
7.1 Om inkrementell systemutvikling i prosjektet.....	61
7.1.1 Inkremitter:.....	61
7.2 Utviklingen av inkremitter.....	62
7.3 Validering og integrering av inkremitter	62
7.4 Vår anvendelse av inkrementell systemutvikling	63
8 Diskusjon av resultater	65
8.1 Resultater	65
8.1.1 Hva ble bra.....	65
8.1.2 Hva ble ikke så bra	66
8.1.3 Ytelseskrav	66
8.1.4 Problemer underveis	66
8.1.5 Hva virker ikke	67

8.2 Alternative løsninger	67
8.2.1 Forslag til forbedringer og nye oppgaver.....	67
8.3 Evaluering av gruppens arbeid.....	67
8.3.1 Innledning	67
8.3.2 Organisering.....	68
8.3.3 Fordeling av arbeid.....	68
8.3.4 Tidsforbruk.....	68
8.3.5 Prosjekt som arbeidsform	68
8.3.6 Subjektiv opplevelse av prosjektet.....	68
8.4 Drøfting av fremdriftsplan	69
9 Konklusjon.....	71
Definisjoner og ordforklaringer	73
Litteratur	75
Vedlegg	77

1 Innledning

1.1 Bakgrunn

Bakgrunn for prosjektet er at vi skal gjennomføre et hovedprosjekt ved Høgskolen i Gjøvik i forbindelse med det avsluttende semester ved studiet Bachelor ingeniørfag, data. Gjennom oppdragsgiver og webansvarlig hos Oppland Arbeiderblad, Torild Akre, ble vi introdusert til deres ønske om en gratulasjonsapplikasjon. Oppgaven skapte interesse blant medlemmene i prosjektgruppen. Vi så prosjektet som en spennende utfordring og noe samtlige gruppemedlemmer var ivrige etter å komme i gang med.

1.2 Målgruppe for rapporten

I utgangspunktet finnes det tre hovedmottagere av denne rapporten. Disse tre er veileder for prosjektoppgaven vår, en ekstern sensor og oppdragsgiver, i vårt tilfelle Oppland Arbeiderblad/A-Pressen Interaktiv. Andre mottakere er studenter ved HiG som skal gjennomføre et eget hovedprosjekt og personer hos Oppland Arbeiderblad/A-Pressen Interaktiv som eventuelt ønsker å videreutvikle løsningen.

Hensikten med denne rapporten er å gi leseren en grundig beskrivelse av hvordan vi har jobbet med prosjektet og hvordan vi har kommet frem til resultatet.

1.3 Formål med prosjektet

1.3.1 Hovedmål

Å lage en gratulasjonsapplikasjon på web for oa.no som ved prosjektets slutt vil være ferdig implementert og fungere tilfredsstillende på OAs server. Applikasjonen skal fungere mot

både nett- og avisutgave og være et effektiviserende hjelpemiddel, både for OA og brukerne.

1.3.2 Prosessmål

Som deltakere i prosjektet ønsker vi å få god innsikt i prosessen med å utvikle et produkt fra idestadiet til en fullverdig applikasjon. Vi tror det vil være nyttig å få erfaring fra å jobbe i et team, spesielt siden dette er en aktuell og fremtidsrettet arbeidsmetode som vi sannsynligvis kommer til å møte i arbeidslivet.

Vi ser det som en utfordring å måtte tilegne oss ny kunnskap om hittil ukjente programmeringsspråk og verktøy. Vi vil i løpet av prosjektet få erfaring med bruk av Microsoft Visual Studio .NET og programmering med C# opp mot en MySQL-database.

Til slutt vil vi nevne at det vil bli en nyttig erfaring å jobbe med en virkelig oppdragsgiver, forholde seg til reelle krav, tidsfrister og avgrensninger.

1.3.3 Effektmål

OA har lenge hatt ønske om å få utviklet et system som vil trekke flere brukere til deres nettsider. Vår løsning vil være et nyttig verktøy for både avisen og dens brukere. I stedet for å måtte ringe inn, eller sende hilsninger manuelt inn til avisen, kan man sende direkte inn til redaksjonen via nettavisen. Både innlegging og betaling foregår via nettet, noe som vil spare OA for den administrative delen. Samtidig vil dette forhåpentligvis også trekke flere folk til nettavisen deres.

1.4 Gruppas arbeidsformer

Vi har gjennom hele prosjektperioden jobbet sammen på vårt tildelte grupperom, rom A115 på HiG. Her har vi brukt personlige datamaskiner under utviklingen av applikasjonen.

Gjennom hele prosjektet har vi hatt faste møter med veileder en gang i uken. På disse møtene har vi holdt veileder oppdatert om prosjektets status og innhentet tips og råd i forbindelse med problemer vi har hatt. I tillegg har vi kontinuerlig skrevet statusrapporter, møtereferater og logger.

Vi har gjennomført flere møter med vår kontaktperson, Torild Akre. Samtidig har mye korrespondanse foregått per e-post.

1.5 Egen bakgrunn og kompetanse

Gruppens medlemmer består av studenter fra dataingeniørlinjen på Høgskolen i Gjøvik, herunder studieretningene drift av datasystemer, systemutvikling og programutvikling. Gruppen har kompetanse innen disse områdene. Ingen av gruppens medlemmer har erfaring med C#.

1.6 Dokumentasjon, konfigurasjon og versjonsstyring

1.6.1 Dokumentasjon

Tekstdokumentene i prosjektet skrives med Microsoft Office 2003. For tegning av usecase-, klasse- og andre diagrammer har vi brukt verktøyet SmartDraw 7.

Vi har brukt følgende tekststandarder i rapporten:

Tekst	Font	Størrelse
Overskrift 1	Arial	24
Overskrift 2	Arial	18
Overskrift 3	Arial	14
Overskrift 4	Georgia (fet/kursiv)	10,5
Normal tekst	Georgia	10,5
Kode	Courier New	9

Tabell 1: Tekststandarder

1.6.2 Konfigurasjon og versjonsstyring

Når gruppa eller et gruppemedlem har jobbet med et dokument skal det lagres med dato og versjonsnummer i filnavnet slik at man lett kan se når dokumentet sist ble oppdatert. Dette for å holde oversikt ved eventuelle tap av nyere versjoner.

1.7 Teknologi

Prosjektet tar for seg flere teknologier som C#/Java-programmering, SSL, CSS, XML og Web Services/SOAP. I tillegg kommuniserer løsningen vår mot en MySQL database ved hjelp av ODBC-klasser. Erfaring med disse teknologiene var relativt liten hos samtlige gruppemedlemmer ved prosjektets oppstart.

1.8 Øvrige roller

Veileder:

Vår veileder gjennom dette prosjektet har vært Rune Lossius. Lossius er høyskolelektor ved HiG og underviser blant annet i www-teknologi. Hans rolle har vært å gi oss tips, råd og veiledning gjennom hele prosjektet.

Oppdragsgiver:

Vår oppdragsgiver er A-Pressen Interaktiv. Vår kontaktperson, Torild Akre, jobber som web-ansvarlig hos Oppland Arbeiderblad som er en av 42 aviser i A-Pressen.

1.9 Organisering av rapporten

Kapittel 1 – Innledning.

Gir en kort innføring i prosjektet og prosjektarbeidet.

Kapittel 2 – Kravspesifikasjon.

Kravspesifikasjonen beskriver alle forhåndsdefinerte krav til systemet.

Kapittel 3 – Analyse og design.

Dette kapitlet gir en grundig beskrivelse av applikasjonen og hvordan vi har valgt å løse oppgaven.

Kapittel 4 – Implementering, koding og produksjon.

Gir en beskrivelse av utviklingsverktøy som er benyttet i prosjektet. Vi beskriver hvordan vi har utviklet de forskjellige inkrementene og kommer med kode-eksempler. I tillegg beskriver vi her de eksterne systemene og hvordan disse er implementert.

Kapittel 5 – Testing og kvalitetssikring.

Beskriver hvordan testing har foregått.

Kapittel 6 – Installasjon

Beskriver hvordan sluttproduktet kan installeres hos andre aviser, og hvilke eksterne ressurser som trengs for å benytte applikasjonen.

Kapittel 7 – Beskrivelse av utviklingsprosessen.

Beskriver hvordan vi har brukt inkrementell utvikling i gjennom prosjektet.

Kapittel 8 – Diskusjon av resultater.

Diskusjon og drøfting rundt produktet, arbeidsmetoder og valg som ble gjort underveis.

Kapittel 9 – Konklusjon

De slutninger vi kom frem til i forrige kapittel.

Kapittel 10 – Litteraturliste

De kilder vi har brukt under prosjektet og en ordliste med tekniske ord og uttrykk som er brukt i rapporten.

Kapittel 11 – Vedlegg

Vedleggene består av møtereferater, fremdriftsplaner, logger, statusrapporter, implementasjonsmanual og annen dokumentasjon.

2 Kravspesifikasjon

2.1 Introduksjon

Kravspesifikasjonen inneholder alle krav vi hadde til systemet før vi startet utviklingen.

2.1.1 Bakgrunn

Oppland Arbeiderblad ønsker et system der nettavisens lesere enkelt skal kunne sende inn bildehilsener til både nettutgaven (oa.no) og papirutgaven. Oppdragsgiver har henvist til vg.no som et slags eksempel på hvordan systemet skal fungere. Forskjellen fra VGs løsning er at den ikke inkluderer muligheten for papirutgavehilsen.

2.1.2 Kort om krav til systemet

Gratulasjonssidene skal bestå av flere sider med mulighet for forskjellig type hilsninger. Eksempler kan være gratulasjonsannonser, våre minste, bryllup osv.

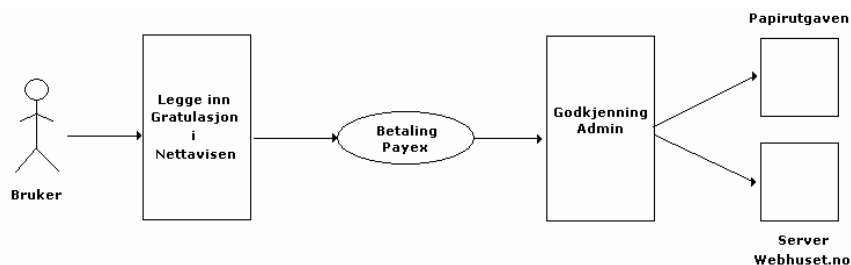
En ansvarlig person hos OA må godkjenne hilsningen før den legges i nettutgaven og oversendes til papirutgaven. Det må med andre ord lages en administrasjonsside i tillegg til gratulasjonssidene. Bildene lastes opp på server der de skaleres for bruk i nett- og papirutgaver. Disse må også fjernes etter en bestemt tid.

Ved bestilling skal annonsen automatisk sendes inn i det redaksjonelle systemet som benyttes i papirutgaven. Her benyttes systemet Saxotech. Pr. i dag er det mulig å sende tekst og bilder direkte inn i det redaksjonelle systemet via e-post.

Som betalingssystem har OA en avtale med Payex. I dag bruker de denne tjenesten til salg av avisen i PDF-format. Gratulasjonsannonsen skal kunne betales direkte på nett ved bestilling. Det blir derfor vår jobb å koble systemet opp mot avisens eksisterende Payex-system. Payex har støtte for betaling via SMS, kredittkort og Payex-konto.

Løsningen skal implementeres som en Web Application. Lagring av alle data skjer mot en MySQL-database.

Vi skal utvikle systemet med mulighet for enkel implementering hos andre aviser da vi kanskje får A-Pressen Interaktiv som oppdragsgiver.



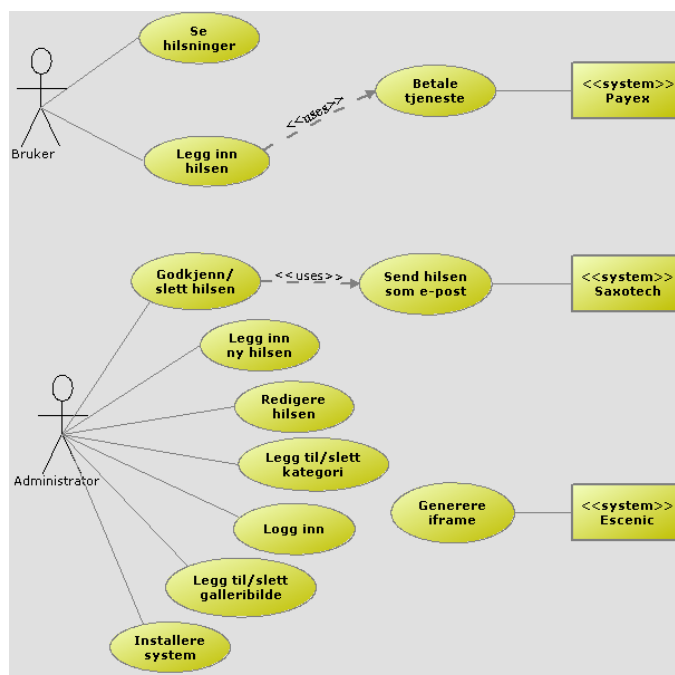
Figur 1: Den overordnede gangen i systemet.

2.1.3 Avgrensinger

Oppdragsgiver har presisert at vi skal prioritere funksjonalitet fremfor design. Det er viktig at systemet fungerer optimalt og er enkelt og effektivt i bruk, både for nettavisens bruker og OAs arbeidere. Systemet skal ligge klart på OAs server hos webhuset.no ved ferdigstilling.

2.2 Usecase

2.2.1 Usecase diagram



Figur 2: Usecase

2.2.2 Usecase beskrivelser

Her følger den formelle beskrivelsen av hvert enkelt usecase.

Usecase: Se hilsninger

Mål: Vise innlagte hilsninger.

Aktør: Bruker

Beskrivelse:

Brukeren vil ha mulighet til å gå inn å se på innlagte hilsninger, både i store og små versjoner. Man vil i først finne en oversikt over forskjellige hilsningskategorier, eksempelvis bursdag, bryllup etc. Inne i hver kategori får man en liste med hilsninger. Disse vises ved et lite bilde, navn på personen det hilses til og dato for hendelsen. Hilsningene er sortert på hendelsesdato. Ved å klikke på en slik hilsen kommer man inn til en større versjon. Her vises all hilsningstekst og det ligger muligheter for utskrift og å tipse venner og bekjente om hilsningen.

Usecase: Legg inn hilsen

Mål: Lage en hilsen for visning på nett/i avis.

Aktør: Bruker

Beskrivelse:

Innlegging av hilsen skjer gjennom en trinn-for-trinn modell.

Prebetingelser:

Bruker ankommer trinn 1 i trinnmodellen gjennom en link i iframe eller menyen til venstre i applikasjonen.

Scenarier:

1. Velger kategori (Bursdag, bryllup, fødsel etc.)
2. Legger inn følgende informasjon om hilsen:
 - Overskrift.
 - Hendelsesdato (Bursdagen, bryllupsdagen etc.).
 - Til navn.
 - Hilsning.
 - Fra navn.
3. Laster opp bilde.
4. Legger inn følgende informasjon om seg selv:
 - Navn.
 - E-post.
 - Telefonnummer.
 - E-postadressen til personer man vil varsle om hilsningen.
5. Velger medium hilsenen skal vises i.
6. Betaler.
7. Mottar bekreftelse på betaling.

Alternative scenarier:

- 2.1. Ett av feltene inneholder feil (antall tegn, format etc.). "Feilmelding".
- 3.1. Bildet er for stort eller er i feil format. "Feilmelding".
- 4.1. Ett av feltene inneholder feil (antall tegn, format etc.). "Feilmelding".
- 6.1. Betaling godkjennes ikke. "Feilmelding: Hilsning ikke Sendt til godkjenning".

Usecase: Godkjenn/slett hilsen

Mål: Godkjenne ferdig hilsen.

Aktør: Administrator.

Beskrivelse:

Her kan administrator godkjenne ferdige hilsninger. Man vil først få en liste over de hilsningene som ikke ennå er godkjent i databasen. Ved å klikke på en link ved siden av hver hilsen, vil man få en forhåndsvisning. Her kan man velge om man vil godkjenne eller slette hilsningen. Ved godkjenning vil hilsning publiseres på nettet og e-poster vil bli sendt til:

- Person som har lagt inn hilsen ("Hilsen er godkjent").
- Saxotech (Hvis hilsen skal inn i papirutgaven.)
- Eventuelle venner og bekjente som er lagt inn med e-postvarsel.

Ved sletting fjernes hilsning (bilder og data i databasen) fra hele systemet.

Usecase: Redigere hilsen

Mål: Endre en allerede innlagt hilsen.

Aktør: Administrator

Beskrivelse:

Her kan administrator kan gå inn å endre på hilsninger som allerede ligger ute på nettet. Man vil først få opp en liste over alle godkjente hilsninger, før man kan gå inn på hver enkelt og endre/lagre dataene for disse.

Usecase: Legg inn ny hilsen

Mål: Legge inn en ny, gratis, hilsen.

Aktør: Administrator

Beskrivelse:

Det vil være mulig for administrator å legge inn en helt ny hilsen. Her skrives data inn på en enkel side. Disse hilsningene vil ikke gå gjennom betalingsmodulen og kan kun vises på web, ikke i papirutgaven.

Usecase: Logg inn

Mål: Logge inn på administratorsiden.

Aktør: Administrator

Beskrivelse:

Administrator logger inn til administratorsidene med brukernavn og passord. Data vil være kryptert. Ved opprettelse av applikasjonen vil disse settes til standardverdier som kan endres av administrator. Standardbrukernavn og passord finner man i medfølgende brukermanual (se vedlegg E).

Usecase: Legg til/slett kategori

Mål: Legge til eller fjerne en kategori.

Aktør: Administrator

Beskrivelse:

Her kan administrator legge til og slette kategorier. Det kan for eksempel være relevant med kategorien "Julehilsen" i desember, men ikke ellers i året. I tillegg til navnet på kategorien må det lastes opp et beskrivende bilde.

Usecase: Legg til/slett galleribilde**Mål:** Legge til eller fjerne et galleribilde.**Aktør:** Administrator**Beskrivelse:**

Hver avis kan ha sitt eget bildegalleri der bruker kan velge bilder dersom vedkommende ikke har et å laste opp selv. Her kan administrator legge til og slette bilder. Det er ikke satt noen fysisk begrensning for antall bilder i galleriet.

Usecase: Installere gratulasjonssystem**Mål:** Installere applikasjonen for nye aviser.**Aktør:** Administrator**Beskrivelse:**

For at en ny avis skal kunne installere gratulasjonssystemet er de nødt til å oppgi en del informasjon. Først må alle filer legges ut på avisens FTP-server. Deretter må administrator gå inn på en installasjonswebseite i vår løsning. Her fyller man inn informasjon om:

- MySQL-database (server, database, brukernavn, passord).
- Avis (Navn, support-mail, URL og logobilde).
- Payex (kontonummer, hash-kode).
- Serverinformasjon (Server-URL, SMTP-server).

Når all informasjon er lagret skal gratulasjonssidene være klare for bruk.

Usecase: Betal tjeneste**Mål:** Gjennomføre betaling.**Aktør:** Administrator**Beskrivelse:**

Betaling foregår ved hjelp av det eksterne systemet Payex. Vi sender kryptert informasjon om pris, kontonummer etc. til Payex og får returnert informasjon om transaksjonens utfall.

Usecase: Send hilsen som e-post**Mål:** Sende hilsning og bilde til avisens redaksjon.**Aktør:** Administrator**Beskrivelse:**

Hilsen sendes som e-post til nettavisens redaksjonelle system, Saxotech. En slik e-post vil inneholde dato, overskrift, selve hilsningsteksten og navn på avsender(e). Bildet sendes som vedlegg.

Usecase: Generer iframe**Mål:** Generere vindu med hilsen for visning på avisens hovedside.**Aktør:** Administrator**Beskrivelse:**

Administrator vil gjennom administrasjonssiden ha mulighet til å endre innholdet i iframen som vises på nettavisens forside. En iframe genereres etter administrators valg. Enten sortert på dato eller med visning av en bestemt hilsen.

2.3 Supplementær kravspesifikasjon

2.3.1 Ytelse

Oppdragsgiver har ikke stilt særskilte krav til systemets ytelse utover at det selvfølgelig skal oppleves som raskt og effektivt. Responstid vil først og fremst være avhengig av brukerens internettilkobling. Deretter kommer serverens hastighet og den tiden denne bruker på å behandle data. Hvis mange brukere er koblet til systemet samtidig kan dette gi redusert responstid. Det sier seg imidlertid selv at antall samtidige brukere ikke vil være skremmende stort da dette er en lokalavis. Vi har ikke satt noen begrensning på hvor mange som kan bruke systemet samtidig.

2.3.2 Sikkerhet

Selve betalingen av tjenester som utføres av systemet foregår gjennom betalingssystemet Payex. Sikkerheten rundt dette tar altså det eksterne systemet seg av. Det skal være mulighet for tilbakebetaling dersom bilde/hilsen ikke blir godkjent av administrator. Hilsninger må godkjennes før de publiseres på internett og i avis.

Administrator logger seg inn på administratorsidene med kryptert passord. Her kan hilsningene behandles, godkjennes eller slettes. Ved innlogging til siden for installasjon/tekniske innstillinger må man også logge inn med et annet brukernavn/passord. Dette for å hindre at uvedkommende roter i disse viktige innstillingene.

2.3.3 Operasjon av systemet

Systemet vil under normal drift kreve en administrator som besøker administratorsidene for godkjenning av bilder med jevne mellomrom. Hvor ofte dette skal skje er avhengig av hvor populær tjenesten blir. Vår anbefaling er at dette skjer en gang om dagen. Det ligger ikke innenfor prosjektgruppens oppgaver å vedlikeholde eller drive nettstedet. Det er heller ikke inngått noen avtale om support eller service på sluttproduktet.

2.3.4 Logging og feilrapportering

Så lenge vår server og kommunikasjonen mot Payex fungerer, skal systemet være feilfritt. Brukere skal få enkle og informative meldinger ved feil som manglende inntasting av navn, e-post er skrevet feil osv. Oppstår andre, store feil, kan systemadministrator kontaktes gjennom support e-post. Supportadresse vil komme til syne på applikasjonens feilmeldingsside dersom det oppstår store

feil i systemet. I tillegg vil supportadresse vises på den siste siden i den trinnvise registreringen i tilfelle bruker har spørsmål i forbindelse med annonsen.

2.3.5 Utvidelser

Det er ønskelig at løsningen skal ha mulighet for videreutvikling i fremtiden. Hvis A-Pressen Interaktiv er interessert i å stå som oppdragsgiver for prosjektet må vi også sette fokus på å få til en flerbrukerløsning slik at man enkelt kan tilpasse produktet til forskjellige nettaviser. Her vil det også være viktig å få til et universelt design som vil passe sammen med flere aviser.

3 Analyse og Design

3.1 Introduksjon

Dette kapitlet vil inneholde informasjon om design og funksjonalitet i applikasjonen vi har utviklet. Vi gir en grov oversikt over design og detaljert informasjon om hva som skjer på sidene våre.

3.1.1 Generelle mål

Vi har hovedsakelig valgt å fokusere på funksjonaliteten systemet skal by på ovenfor brukeren. Dette vil si den trinnvise registreringen av hilsninger og muligheten til å navigere mellom og se på innlagte hilsninger. Målet vil være på å lage en tjeneste som er enkel og attraktiv å bruke for brukeren.

I administrasjonsdelen har vi også lagt vekt på funksjonalitet og feilsjekking. Vi ser det som naturlig at avisens ansatte har noe høyere kunnskap om løsningen enn den vanlige bruker og har derfor ikke fokusert like mye på hjelpeverktøy som i brukerdelen.

3.2 Funksjonelt design

Funksjonalitet henger sammen med brukervennlighet. Det er for eksempel viktig at navigasjonsmeny har samme gjennomgående plassering på samtlige sider. Dette for at brukeren skal ha viktige sider tilgjengelig hele tiden, samme hvor vedkommende beveger seg i applikasjonen.

Siden VGs gratulasjonssider er angitt som et eksempel på hvordan løsningen kan utvikles, vil vi prøve å forbedre det vi ser på som svakheter i deres løsning. Vi har søkt hjelp til dette gjennom en uformell spørreundersøkelse der vi prøver å kartlegge hvordan brukerne oppfatter VGs funksjonalitet, hva som eventuelt kan forbedres osv. Gjennom denne undersøkelsen var det noen punkter som gikk igjen hos testkandidatene:

- Overflødige og lite oversiktlige menyer. Store ikoner og store avstander.
- Ukonsekvent bruk av ”neste” eller ”gå videre”.
- Hva med en side med hilsninger sortert etter dato!?
- Dårlig bildeutvalg i databasen (hvis man ikke har eget bilde).
- Dårlig sikkerhet. Mulig å laste opp bilder i tekstboksene ved hjelp av script osv.

Vi vil ta hensyn til disse punktene i det videre prosjektarbeidet og prøve å unngå ”feilene” VG har gjort.

3.3 Databasedesign

Vi har ingen komplisert databasestruktur. Under finnes en kort beskrivelse av hver enkelt tabell.

The image shows a screenshot of a database design tool with seven table definition windows. Each window lists the table name and its columns with data types and constraints.

- avis**:
 - bilde_logo: VARCHAR(50)
 - epost_saxotech: VARCHAR(50)
 - epost_smtpserver: VARCHAR(50)
 - epost_support: VARCHAR(50)
 - navn_avis: VARCHAR(50)
 - payex_hash: VARCHAR(30)
 - payex_kontonr: INTEGER(8)
 - payex_url: VARCHAR(50)
 - url_avis: VARCHAR(50)
 - url_losning: VARCHAR(50)
 - brukernavn: VARCHAR(30)
 - password: VARCHAR(255)
- pris**:
 - mediumtype: VARCHAR(10)
 - pris: CHAR(3)
- sortering**:
 - sortstring: VARCHAR(200)
 - visning_id: VARCHAR(6)
- galleri**:
 - id: INTEGER(6)
 - bildepath: VARCHAR(50)
 - thumbpath: VARCHAR(50)
- kategori**:
 - navn: VARCHAR(30)
 - bilde: VARCHAR(30)
- hilsen**:
 - hilsen_id: INTEGER(11)
 - overskrift: VARCHAR(50)
 - hilsen_til: VARCHAR(50)
 - hendelsesdato: DATE
 - fradato: DATE
 - hilsen: VARCHAR(150)
 - hilsen_fra: VARCHAR(50))
 - fulltnavn: VARCHAR(150)
 - epost: VARCHAR(50)
 - telefon: INTEGER(8)
 - filpathbilde: VARCHAR(100)
 - filpaththumb: VARCHAR(100)
 - kategori: VARCHAR(30)
 - medium: VARCHAR(10)
 - godkjent: INTEGER(2)
 - epostvarsling: VARCHAR(255)
 - transNr: INTEGER(11)
- tmp_hilsen**:
 - tmp_hilsen_id: INTEGER(11)
 - tmp_overskrift: VARCHAR(50)
 - tmp_hilsen_til: VARCHAR(50)
 - tmp_hendelsesdato: DATE
 - tmp_fradato: DATE
 - tmp_hilsen: VARCHAR(150)
 - tmp_hilsen_fra: VARCHAR(50)
 - tmp_fulltnavn: VARCHAR(50)
 - tmp_epost: VARCHAR(50)
 - tmp_telefon: INTEGER(8)
 - tmp_filpathbilde: VARCHAR(100)
 - tmp_filpaththumb: VARCHAR(100)
 - tmp_kategori: VARCHAR(20)
 - tmp_medium: VARCHAR(10)
 - tmp_godkjent: INTEGER(2)
 - tmp_epostvarsling: VARCHAR(255)
 - tmp_transNr: INTEGER(11)

Figur 3: Database

Avis

Inneholder informasjon om avisen. Denne informasjonen fylles ut gjennom installasjonsdelen.

Sortering

Her lagres en sorteringsstring. Denne beskriver hva som skal vises i iframen.

Hilsen

Informasjon om hilsninger som er betalt og ligger klar for godkjenning eller er godkjent.

Grunnen til at vi har valgt å ha to like tabeller (hilsen og tmp_hilsen) er at dette gjør det enklere å holde orden i/slette uferdige og ferdige hilsninger. Når en hilsen er ferdig og betalt flytter vi den fra tmp_hilsen til hilsen.

Tmp_hilsen

Informasjon om hilsninger som ennå ikke er betalt, uferdige hilsninger osv.

Kategori

Inneholder navn og bilde for hilsningskategorier. Vi lagrer ikke selve bildet, men en link til stedet der bildet ligger på serveren.

Galleri

Inneholder galleribilder. Disse er lagret på samme måte som bildene i kategoritabellen.

Pris

Inneholder informasjon om priser på de forskjellige tjenestene.

3.4 Datastruktur

Visual Studio .NET har et veldig oversiktlig og greit oppsett for bruk av klasser. I vår kode har vi laget en rekke egne klasser og fått en ferdig fra Payex. Under har vi oppsummert funksjonaliteten i klassene.

Bilde

Inneholder funksjoner som har med bildebehandling å gjøre. Interpolasjon, sjekk av bildestørrelse og format etc.

CryptFil

Inneholder funksjoner for kryptering av innloggingspassord, tilkoblingsstring etc.

E-post

Brukes for sending av e-post, en operasjon som går igjen flere steder i applikasjonen. Vi sender e-post til hilsningens avsender, mottaker, til avisredaksjonen etc.

Tekst

Inneholder enkle funksjoner. Et eksempel er en funksjon som gjør første bokstaven i et ord/setning stor.

Database

Dette er vår største klasse. Denne inneholder alle funksjoner som kommuniserer mot MySQL-databasen. Her har vi blant annet laget to generelle funksjoner, SkrivTilDB og HentFraDB, som brukes veldig mye gjennom hele applikasjonen.

HTML-tabell

Klasse med funksjoner som genererer ulike HTML-tabeller.

MD5Hash

Klasse med funksjoner for kryptering av data som sendes mellom vår applikasjon og Payex.

Parseresult

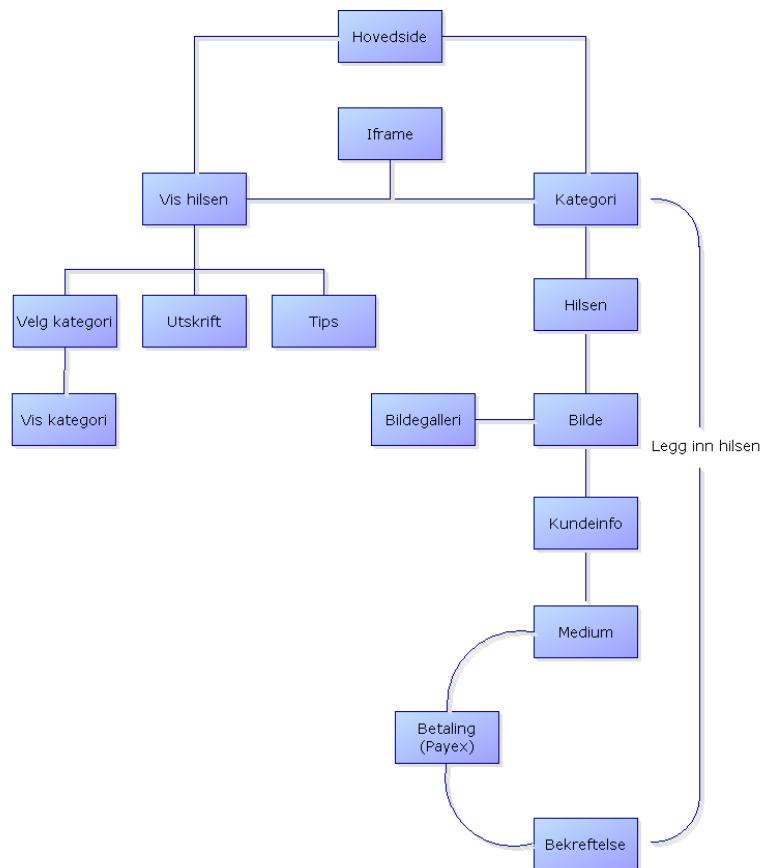
Klasse som inneholder funksjon for XML-parsing.

PxOrder

Ferdig klasse vi har mottatt fra Payex. Brukes for initiering og bekreftelse av betaling.

3.5 Funksjonell struktur - brukerdel

3.5.1 Navigasjonskart



Figur 4: Navigasjonskart for brukerdelen

Figuren på forrige side viser en oversikt over hvordan strukturen på nettsiden ser ut slik nettavisens brukere vil oppleve den. Den lange rekken til høyre viser registreringsdelen hvor bruker legger inn en hilsning i systemet. Her er Payex et eksternt system som tar seg av betalingen.

3.5.2 Sidene

Iframe

Vår "iframe" blir satt inn i et lite vindu (HTML-iframe) på nettavisens forside. Dette er med andre ord vårt ansikt utad og entringspunktet for de fleste brukere. Vinduet viser et bilde av en utvalgt hilsning samt en link, "Legg inn hilsen". Vinduet skal være 150 piksler bredt. Ved å klikke på bildet vil man sendes direkte til den viste hilsningen. Klikker man på "Legg inn hilsen" sendes man til trinnvis registrering av ny hilsen. Hilsning som vises i iframe kan spesifiseres av administrator. Her vil det finnes 2 valgmuligheter:

Valg 1: Hvis ikke annet er angitt vil nyeste hilsning, sortert etter hendelsesdato (dato vedkommende har bursdag, bryllupsdag etc.) vises i iframe. Finnes det flere med hendelsesdato i dag, vil disse vises i tilfeldig.

Valg 2: Den andre muligheten er at administrator kan velge en spesifikk hilsen for visning i en dag. Denne vil fjernes igjen automatisk når dagen er over og visning endres tilbake til valg 1.

I tillegg inneholder vår iframe informasjon om tjenesten i form av en kort, forklarende tekst.



Figur 5: Iframe

Vis hilsen

Denne siden viser den store versjonen av hver hilsen. Her finner man all informasjon som er lagt inn om hilsningen. Under hilsningen ligger knapper med mulighet for "utskriftsvennlig side" "tips en venn", og "gå tilbake".

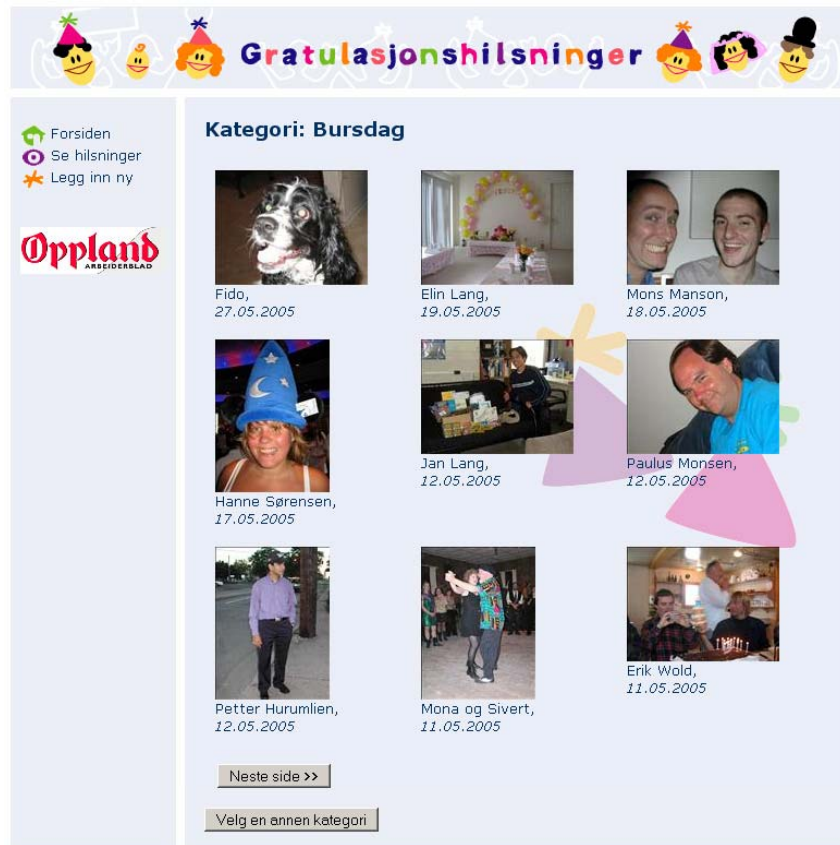


Figur 6: Visning av hilsen.

Siden åpnes ved at bruker velger en spesifikk hilsen, enten fra "iframe" eller "vis kategori". Hilsnings-ID for hilsen som skal vises sendes med som en querystring i adressefeltet i nettleseren.

Vis kategori

Hver kategori har sin egen side der det vises bilder med navn og dato. Disse er sortert kronologisk etter hendelsesdato. På hver side vil det legges totalt ni hilsninger. Når dette antallet overstiges genereres en ny side med plass til ni nye hilsninger. Antall hilsninger per side er valgt for å unngå for mye scrolling i skjermbildet skjermbildet. Hilsningene vil ligge ute i 31 dager før de slettes automatisk. Ved å klikke på en liten hilsning får man opp den store versjonen.



Figur 7: Visning av samtlige hilsninger i en kategori.

Tips

Et "pop-up-vindu" der man kan taste inn e-post adresse til en person man ønsker å tipse om en utlagt hilsning. Her tastes følgende info inn:

- Mottakers e-post.
- Senders e-post.
- Kommentar.

Tipset sendes umiddelbart etter at bruker har klikket "Send". Vinduet lukkes automatisk når hilsningen er sendt.

Utskriftsvennlig side

Et "pop-up vindu" der man får opp hilsningen på nytt, tilpasset på et A4-ark. Samtidig startes utskriftsmenyen for direkte utskrift.

Legg inn hilsen

Innlegging av hilsen kan startes direkte fra nettavisens forside (iframe) og fra samtlige sider i vår applikasjon gjennom menyen til venstre. Bruker blir veiledet gjennom en 7-trinns registrering av hilsningen. Til venstre i menyen vil det under registreringene være

en "trinn-modell" der bruker til enhver tid holdes orientert om hvor han er i registreringen.

For å hjelpe brukeren har vi plassert hjelpeikoner der det er nødvendig. Ved hjelp av disse kan bruker få opp små hjelpebokser i tilfelle det ikke er klart hva som skal fylles inn.

I samtlige innfyllingstekstbokser sjekkes følgende:

- At nødvendige felter er utfylte.
- At man ikke fyller inn script eller HTML-kode.
- At tekst (f.eks. telefonnummer og e-postadresse) har rett format.

Nedenfor følger en mer detaljert beskrivelse av hvert enkelt trinn.

1) Kategori

I første trinn velger bruker hvilken kategori han vil opprette en hilsen i. Kategorivalget sendes med til trinn 2 som en såkalt "querystring" i adressefeltet på nettleseren.



Figur 8: Førstesiden hvis man velger å legge inn ny hilsen.

2) Hilsen

Her starter innhenting av informasjon fra bruker. På denne siden skriver bruker inn:

- Overskrift (30 tegn).
- Hendelsesdato (dato for bursdag, bryllup etc.)
- Hilsen til (Navn på hilsenens mottaker, 130 tegn)
- Hilsen (Selve hilsenen, 120 tegn).
- Hilsen fra (Navnet på person(er) som hilser, 30 tegn).

Feilsjekker:

- At dato er nyere eller lik dagens dato.
- At dato er fra og med dagens dato og 25 dager frem i tid.

Hendelsesdato vil være datoen hilsningen vises i iframe på nettavisens forside. Hilsningen vil vises på nettet fra og med dagen den blir godkjent. Før bruker går videre opprettes en cookie som brukes til å identifisere hilsningen. Denne vil bli brukt når nye data legges til i de kommende trinnene. Når bruker velger å gå videre lagres alle inntastede data, samt hilsningens kategori med hilsningens ID i en temporær database. Til slutt settes hilsningen til "ikke godkjent".

Figur 9: Lag hilsen.

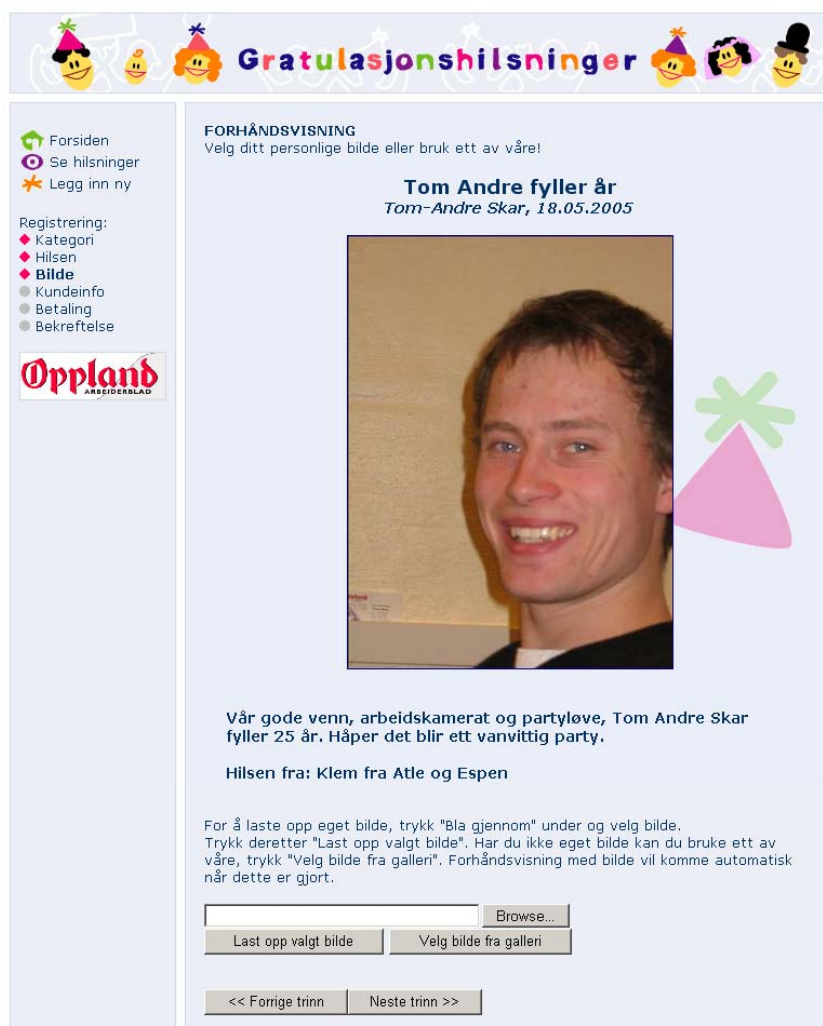
3) Bilde

En side der bruker kan velge å laste opp bilde eller bruke et av avisens bilder som kan hentes fra et galleri. Ved å trykke "Bla Gjennom..." vil bruker få opp et vanlig browservindu der man kan klikke seg frem til ønsket bilde. Bruker trykker så "Last opp valgt bilde" og får automatisk opp bilde i forhåndsvisning sammen med øvrig lagret tekst. Om bildet hentes fra galleriet, vil det automatisk vises sammen med øvrig tekst når man kommer tilbake til denne siden.

Feilsjekker:

- At bilde har JPG/JPEG-format.
- At bilde ikke er større en 2MB.
- Bilde *må* være lastet opp før man får lov å gå videre.

Bildene lagres på server som en stor og en liten bildefil, mens deres filadresse lagres i databasen.



Figur 10: Opplasting av bilde med forhåndsvisning.

4) Brukerinfo

Informasjonen om avsender av hilsen legges inn på denne siden slik at administrator kan kontakte vedkommende om hilsningen ikke blir godkjent. I tillegg er det her være mulig å legge inn e-postadresser for varsling av familie, venner og bekjente av personen det hilses til. For at e-postadressene skal leses korrekt av vår applikasjon er vi avhengig av at e-postvarslingsboks nummer 1 fylles ut først. For å løse dette har vi deaktivert de andre innfyllingsboksene inntil boks nummer 1 er utfyllt.

Følgende info tastes inn:

- Fullt navn
- E-post
- Telefon
- E-postvarsel (til venner og bekjente).

All informasjon lagres i databasen, under samme ID som i forrige trinn ved hjelp av cookie.

Figur 11: Innfylling av brukerinformasjon.

5) Medium

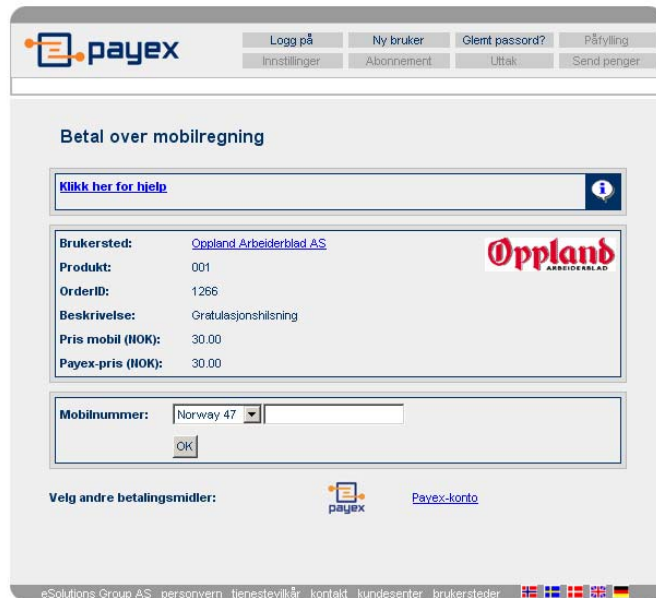
Til slutt velger man hvilke medium annonsen skal vises i. Man kan velge visning i avis, på web eller i begge deler. Prisen for de forskjellige mulighetene vises til brukeren i det han gjør valget. Avhengig av hvilket medium som er valgt sendes brukeren videre til Payexs sider for fullføring av transaksjon. I vår løsning er det kun hilsninger i bursdags-, fødsels- og bryllups-kategorien som kan sendes inn til avisen. De andre vises bare på nett. Før man starter betalingen lagres medium i databasen.

Figur 12: Valg av medium.

6) Betaling

Betalingen gjennomføres ved hjelp av Payex. Bruker videresendes til Payexs hjemmesider for gjennomføring av transaksjon. En hilsen kan betales med Payex-konto, SMS og kredittkort. Betalingsmiddel velger man nederst på Payex sin side.

I valg av design på våre sider har vi forsøkt å lage noe som ikke er veldig ulikt Payex sitt. Dette for at bruker skal føle at han opererer i ett og samme system under hele registreringen.



Figur 13: Payex sin betalingside.

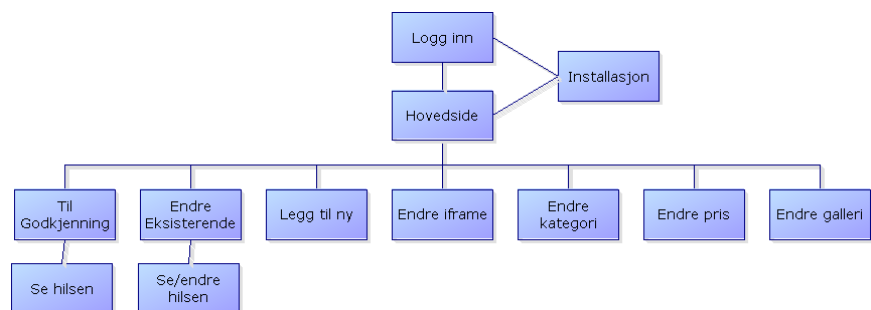
7) Bekreftelse

Ved fullført betaling dirigeres brukeren tilbake til en statusside. Her gis melding om status for betaling samt referansenummer hos Payex. Hvis betalingen er godkjent flyttes hilsning over i hoveddatabasen, der den vil ligge inntil den blir godkjent av administrator.

Hvis det har oppstått feil ved betalingen og denne ikke blir godkjent, gis brukeren mulighet til å kontakte support på denne siden.

3.6 Funksjonell struktur - administratordel

3.6.1 Navigasjonskart



Figur 14: Navigasjonskart for administratordelen.

Navigasjonskartet over viser strukturen på administrasjonssiden sett fra administrators ståsted. Vi har gjort de fleste funksjonalitetene tilgjengelige fra hovedsiden. Det er bare

godkjenning/redigering av hilsninger man utfører på sidene lenger ned i navigasjonstreet.

3.6.2 Sidene

Logg inn

Brukernavn og passord kontrolleres mot databasen. I databasen ligger passordet kryptert. Ved korrekt innlogging sendes man til administratorsiden. Her har man tilgang i 45 minutter fra innlogging før cookie slettes og man må logge inn på nytt.

Hovedsiden

Dette er siden som møter deg når du logger deg inn som administrator. Her har vi samlet alle funksjonalitetene administrator har bruk for under daglig drift av systemet.

Den øverste delen på siden dreier seg om hilsninger og vil alltid være synlig for brukeren. Her bør administrator gå inn å sjekke hilsninger som ligger "til godkjenning" så ofte som mulig. Man kan også endre eksisterende hilsninger, eller legge til en helt ny.

I den nederste delen kan man velge innhold fra menyen til venstre. Her kan man administrere følgende:

- Iframe.
- Pris.
- Kategori.
- Galleri.
- Brukernavn/passord.
- Tekniske innstillinger.

Gratulasjonshilsninger	
Administrator	
Meny: - Endre iframe - Endre kategori - Endre pris - Endre galleri - Endre teknisk - Endre Passord og Brukernavn	Hilsninger: Velkommen til administrasjonssiden. Her kan du godkjenne, endre og slette hilsninger, samt utføre andre endringer i applikasjonen. Vær forsiktig og les nøye gjennom hjelpen som finnes for hvert verktøy før du lagrer. Til godkjenning (1stk.) Endre eksisterende (13stk.) Legg til ny

Figur 15: Administratorsidens forside.

Endre iframe

I denne delen kan administrator bestemme hva som skal vises i på nettavisens hovedside (iframe). Her finnes to forskjellige valg.

1. Ved sortering på dato vil den hilsen med hendelsesdato nærmest opp til, eller lik, dagens dato. Hvis flere hilsninger har dagens dato vises disse tilfeldig hver gang iframe oppdateres.
2. Velger man derimot "Vis ID" vil man få opp en boks for utfylling av ID til hilsningen man vil vise. Hilsningen som velges må ha hendelsesdato lik dagens dato, hvis ikke kommer det feilmelding om dette. Når en bestemt hilsen er valgt i iframe vil den vises her i en dag før det automatisk tilbakestilles til sortering pr. dato.

Endre pris

I denne delen ligger prisen på de ulike tjenestene. Her kan administrator sette prisen på å ha hilsningen i avis, på nett eller i begge medier.

Endre kategori

Her kan man endre og slette de forskjellige kategoriene. Dette fordi enkelte kategorier kanskje bare er aktuelle enkelte perioder i året. Til hver kategori må det lastes opp et beskrivende bilde. Dette bilde skaleres automatisk. Likevel anbefales det at alle bilder til dette formål er i samme størrelse fordi sidene da blir penest å se på. Ved installering finnes det 6 hilsningskategorier, bursdag, bryllup, fødsel, jubileum, en hilsen fra meg og kjærlighet.

Endre galleri




























Avisen har et eget galleri der brukeren kan velge bilde om vedkommende ikke har et eget å laste opp. Også her får bildet automatisk riktig størrelse når det lastes opp. Vi anbefaler fremdeles å laste opp bilde i samme størrelse. Det er ingen begrensning på hvor mange bilder man kan legge i galleriet.

Endre brukernavn og passord

Her kan passord og brukernavn endres. Vi anbefaler at dette gjøres da disse settes til en standard ved installering. For å legge inn nytt brukernavn/passord må man bekrefte ved å skrive inn gammelt passord.

Gratulasjonshilsninger

Administrator

Meny: - Endre iframe - Endre kategori - Endre pris - Endre galleri - Endre teknisk - Endre Passord og Brukernavn	Hilsninger: Velkommen til administrasjonssiden. Her kan du godkjenne, endre og slette hilsninger, samt utføre andre endringer i applikasjonen. Vær forsiktig og les nøye gjennom hjelpen som finnes for hvert verktøy før du lagrer. Til godkjenning (1stk.) Endre eksisterende (13stk.) Legg til ny									
	Galleri, Legg til og slett bilder Her kan du legge til og slette bilder fra bildegalleriet. Herfra kan brukeren finne alternative bilder dersom han ikke har ett eget å laste opp. Last opp nytt: <input style="width: 100px;" type="text"/> <input type="button" value="Browse..."/> <input type="button" value="Lagre galleribilde"/>									
	Slett: <input type="button" value="Slett merket galleribilde"/>									
	<table style="width: 100%; text-align: center;"> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>									
										
										
										

Figur 16: Verktøy for å legge til/fjerne galleribilder.

Godkjenn hilsen

På denne siden kommer en liste over hilsninger som ikke er godkjent opp. Her vil vi også vise hilsningens:

- Id.
- Dato
- Kategori
- Navn/e-post
- Telefonnummer.
- Medium.
- Transaksjonsnummer.

Dette for å gjøre det enkelt for administrator å kontakte de som har sendt inn og betalt for hilsningen dersom den ikke blir godkjent. Hilsningene sorteres automatisk på ID, men kan også sorteres etter de andre attributtene ved hjelp av sorteringsverktøyet øverst til venstre.

Ved å trykke "Se hilsen" vil man få en enkel forhåndsvisning av hilsningen slik at man kan vurdere dens innhold. Deretter kan man velge om man vil godkjenne eller slette den.

Ved godkjenning settes attributten til godkjent i databasen og det sendes e-post til:


- Bruker (om at hilsningen er godkjent).
- Saxotech (hvis hilsningen skal inn i avisen).
- Eventuell familie og venner som skal motta e-postvarsel.

Ved sletting fjernes all hilsningsdata og bilder fra database og server.

Forhåndsvisning

Overskrift:
Olivia og Arne together 4ever

Hilsen til, Dato:
Olivia og Arne, 18.05.2005



Hilsen:
Hurra for vårt lykkelige par.

Hilsen fra:
Fredrik Lane

Tilbake Godkjenn Slett

Figur 17: Forhåndsvisning for godkjenning av hilsen.

Endre eksisterende hilsninger

På denne siden kommer en liste over hilsninger som allerede er godkjent og ligger ute på nettet. Hilsningene er sortert på ID, men

kan også sorteres etter andre attributter med funksjonen øverst til venstre.

Her vises de som er godkjent:

Sorter etter:
ID

ID	Dato	Kategori	Navn, Epost	Telefon	Medium	Trans.Nr.	Se/endre hilsen
155	19.04.2005	kjærlighet	Hanne Sørensen, Hanne.Sorensen@stud.nhh.no	97512840	web	1045813	Se/endre hilsen
158	22.04.2005	en hilsen fra meg	Mari Grafsrønningen, magrafs@online.no	97135741	web	1045816	Se/endre hilsen
163	22.04.2005	en hilsen fra meg	Magne Hoseth, magneh@online.dom	53434323	web	1045893	Se/endre hilsen
166	25.04.2005	en hilsen fra meg	Espen B. Mikalsen, ebmikalsen@hotmail.com	99030302	web	1045960	Se/endre hilsen
167	17.05.2005	bursdag	Atle G., shanahan18@hotmail.com	93223297	avisweb		Se/endre hilsen
168	25.04.2005	bursdag	Hanne Sørensen, hanne.sorensen@hotmail.com	97512850	web		Se/endre hilsen
170	03.05.2005	en hilsen fra meg	Espen B. Mikalsen, ebmikalsen@hotmail.com	99030302	web		Se/endre hilsen
174	05.05.2005	bursdag	Fia Lien, lien@fina.com	12345678	web		Se/endre hilsen
177	04.05.2005	bursdag	Mamma Nilsen, mamma@nilsen.net	88776655	web		Se/endre hilsen
181	19.05.2005	bursdag	Olav Horde, adoptiv@org.net	11224455	web		Se/endre hilsen
185	18.05.2005	bursdag	Gary Olafsen, gary@dott.nett	99887766	web		Se/endre hilsen
186	04.05.2005	bursdag	Teng tang, atle.gulbrandsen@hig.no	11223344	web		Se/endre hilsen
187	27.05.2005	bursdag	Frøydis Sæther, fro@di.do	9231123	web		Se/endre hilsen

Tilbake

Figur 18: Liste over eksisterende hilsninger.

Ved å trykke "se/endre hilsen" kommer man inn på en forhåndsvisning der man kan endre alle attributtene til hilsningen. Deretter trykker man "Lagre" for å legge ut hilsningen med nye data.

Forhåndsvisning

Vil sende deg en liten hilsen

Tora 03.05.2005



Hei Tora! Ville bare sende deg en liten hilsning for å si at jeg er veldig glad i deg og håper du har det bra. Hils til mamma, pappa og bror.

Espen

Tilbake Slett Lagre endringer

Figur 19: Her kan man endre hilsningsdata.

Legg inn ny hilsen

Her kan administrator lage en ny hilsen og legge den direkte ut på nettet. Hilsningene som lages her vil ikke gå gjennom

betalingsdelen. Det er *kun* mulig å legge hilsningene ut på nett, ikke i avisen.

Installasjon/Tekniske innstillinger

Dette er installasjonssiden for vår løsning. All informasjon på denne siden må fylles ut og lagres før løsningen kan tas i bruk av en ny avis. Informasjonen kan også endres senere ved å velge "Tekniske innstillinger" på administrators hovedside.

Oppsettet består av fire deler. Det er viktig at informasjonen fylles inn og lagres i nummerert rekkefølge. Dette fordi forbindelsen mot database må være i orden før noe av informasjonen i de andre punktene kan lagres.

1. Her fyller man inn server, database, brukernavn og passord. Dette er informasjon avisen vil få fra sin serverleverandør. I det man klikker "Lagre" vil tilkoblingen testes før den opprettes.
2. Her fyller man inn informasjon om avisen. Avisens navn, adressen til avisens hjemmeside, e-postadresse for support og opplasting av bilde for bruk til logo. Logoen vil bli vist i menyen til venstre i nettleseren. Informasjonen vil bli lagret til databasen i det man trykker "Lagre".
3. I denne delen fyller man inn informasjon om serveren. Her trenger vi server URL og SMTP-server. URLen er adressen der løsningen ligger. For eksempel www.oa.datasenter.no. SMTP-serveren vil man få oppgitt hos serverleverandør. Denne trengs for at man skal kunne sende e-post fra serveren.
4. I den siste delen fyller man inn informasjon om Payex. Først avisens kontonummer hos Payex og deretter hashkoden. Hashkoden får man ved henvendelse til Payex. Her kan man også velge om man vil kjøre applikasjonen mot Payex testmiljø, eller mot produksjonsmiljø. I produksjonsmiljøet foretas reelle betalinger, mens man i testmiljøet bare tester applikasjonen mot Payex med fiktive betalinger.

Tilleggsfunksjonalitet

I tillegg til funksjonaliteten vi har nevnt så langt ligger det to andre funksjoner skjult i systemet. Disse kjøres en gang, hver gang administrator logger inn på siden.

1. Temporær database reduseres til 15 poster. Alle poster utenom de 15 nyeste slettes. Vi har valgt å la det være igjen 15 stykker i tilfelle noen legger inn hilsen samtidig som slettingen gjøres. Da unngår vi at dataene til vedkommende som legger inn slettes.
2. Sletter hilsninger som er eldre enn 31 dager. Går gjennom hilsener som ligger ute og sletter de der det har gått mer enn en måned siden innleggelsesdato. Hilsninger skal i utgangspunktet ligge ute i en måned. Vi har likevel valgt å legge til en ekstra dag siden det normalt kan ta litt tid før hilsning godkjennes av administrator.

4 Implementering, koding og produksjon

4.1 Utviklingsmiljø

I utgangspunktet hadde vi håpet at skolen kunne holde oss med PC-er. Dette viste seg fort å være vanskelig da maskinene skolen hadde til disposisjon var av for dårlig kvalitet for vårt bruk. Det hele endte dermed med at hele gruppen stilte med egne bærbar PC-er og installerte nødvendig programvare på disse.

4.2 Programvare

Programvare benyttet i forbindelse med prosjektet:

- Smart FTP.
- SmartDraw 7.
- Microsoft Word 2003.
- Microsoft Visual Studio .NET 2003.
- Adobe Acrobat Professional.
- Microsoft Internet Explorer 6.
- Mozilla Firefox.
- MySQL.
- MySQL-Front.
- MyODBC-drivere.
- MicroOLAP Databasedesigner.
- ODBC Driver.

4.3 Hardware

Hvis dette blir et flerbrukersystem trenger hver avis som skal ta det i bruk en ekstern server. Denne serveren bør støtte samme funksjonalitet som vår server hos webhuset.no. Følgende spesifikasjon er gitt for vårt webhotell på webhuset.no

- Webhotell, størrelse MIDI.

- Månedspris 99,- (+49,- pr mndfor MySQL-database).
- Diskplass 100MB.
- POP3-kontoer: 10.
- Trafikk: 10GB pr mnd.
- SSL-støtte (+ 998,- pr år).

4.4 Valg av verktøy

4.4.1 Programmeringsverktøy

Microsoft Visual Studio .NET

Oppdragsgiver hadde ingen føringer med tanke på kode og utviklingsverktøy. Valget falt derfor på C# og .NET, etter samtale med veileder. I tillegg er dette et verktøy som er "i tiden" og kan være flott å ha erfaring fra senere.

4.4.2 Kodestandard

Det er ikke fulgt noen spesiell standard under utviklingen, men vi har fulgt de samme retningslinjene innad i gruppen som vi avtalte på forhånd. Dette er stort sett standarder vi har opparbeidet oss under fagene Grunnleggende programmering og Objektorientert programmering. På denne måten har vi fulgt vår egen standard gjennom hele prosessen.

Under utviklingen har vi prøvd å ha så gode rutiner som mulig på kommentering av kode. Dette for at utenforstående enkelt skal kunne sette seg inn i koden.

Navnsetting på funksjoner og variable har også vært viktig. Dette for å gi gruppens medlemmer lik forutsetning for å forstå koden raskest mulig. Noen eksempler på navnsetting vi har brukt:

- txtFornavn
- lblFornavn

Tekstbokser begynner alltid "txt" før en forklarende tekst med stor første bokstav. (Labels starter med lbl osv.) Alle funksjoner starter på stor bokstav med forklarende navn, slik at vi lett skiller disse fra variable.

Ved valg av språk under utviklingen valgte vi å kode og kommentere så godt det lot seg gjøre på norsk. Utviklingsverktøyet vårt, VS .NET, var jo engelsk, så mange av de funksjonene og klassene som vi ikke hadde lagd selv ble på engelsk. Kildekode som kom i fra av Payex var på engelsk. Vi så ingen grunn til å bruke tid på å oversette dette.

Under selve utviklingen har vi stort sett sittet hver for oss. Når noen stod fast benyttet vi oss av par-programmering. Dette ble en stor suksess da vi opplevde at problemene løste seg mye fortere. Man kan ofte se seg blind på egen kode og da er det effektivt å

hente inn hjelp fra andre. Her har også innhenting av informasjon fra nettet vært til god hjelp.

4.4.3 Kommunikasjon mot database

Vår første oppgave var å få satt opp utviklingsmiljøet. Planen var fra begynnelsen å benytte eksisterende MSSQL-2000 database, som ligger ferdig implementert i VS .NET. Dette viste seg derimot umulig da OAs server kun støttet MySQL.

Etter en del prøving og feiling var MySQL-databasen på plass og vi kunne ta i bruk ODBC-klassene for kommunikasjon mellom systemet og databasen. ODBC er et standard interface som gjør at alle databaser er like sett fra utsiden.

4.4.4 Kode eksempler fra Database.cs

```
using System.Data.Odbc;

//Skriver gitt data til "feltNavn" i gitt SQL-string.
public static void SkrivTilDB(string sqlString)
{
    //Tilkobling() returnerer tilkoblings string
    OdbcConnection con = new
    OdbcConnection(Tilkobling());

    //Åpner tilkoblingen
    con.Open();

    //SQL-string er SQL-kommandoen vi kjører mot
    databasen.
    OdbcCommand cmd = new OdbcCommand(sqlString, con);

    //Utfører SQL kommandoen.
    cmd.ExecuteNonQuery();

    //Avslutter tilkoblingen.
    con.Close();
}

//Funksjon som fyller "dropdownlistene" våre.
public static void FyllDropDownList
(System.Web.UI.WebControls.DropDownList ddl)
{
    OdbcConnection con = new
    OdbcConnection(Tilkobling());

    //Initsierer en dataleser
    OdbcDataReader datareader;

    //Ny SQL-kommando
    OdbcCommand cmd = new OdbcCommand("SELECT navn FROM
    kategori", con);

    //Åpner tilkoblingen
    con.Open();

    datareader =
    cmd.ExecuteReader(CommandBehavior.CloseConnection);
```

```

//Så lenge det eksisterer data i databasen vil
dataleseren returnere innholdet i posten og legge
det til i vår "dropdownlist".
while(datareader.Read())
{
    ddl.Items.Add
    (datareader["navn"].ToString());
}

//Avslutter tilkoblingen.
con.Close();
}

```

4.5 Cookies

Det finnes flere muligheter for å sende informasjon mellom sider. Den enkleste vi vurderte var en såkalt querystring. Et lite minus ved denne metoden, er at verdien som sendes vises i nettleserens adressevindu. Dette gjør at brukeren kan se og endre hvilken hilsen-ID det jobbes mot. Vi har valgt å bruke querystring enkelte steder i applikasjonen. Dette er steder der vi føler verdien som sendes ikke inneholder sensitiv informasjon. For eksempel ved sending av kategori mellom første og andre trinn i "Legg inn hilsen". En fordel ved bruk av querystring er at brukeren kan bokmerke siden hvis han har lyst til å gå tilbake til hilsningen ved en senere anledning.

Valget av metode for sending av ID falt til slutt på cookies. En ulempe med denne metoden er at cookies kan være deaktivert i nettleseren. Ettersom Payex bruker cookies og nettleseren må akseptere disse for å gjennomføre betalingen, valgte vi også å bruke denne metoden. Vi har derfor laget en advarsel om bruken av cookies på vår forside. Der forklarer vi også hvordan man kan aktivere cookies i nettleseren.

Når registreringen av en ny hilsning blir påbegynt blir det opprettet en ny post med unik ID i den temporære databasen. Dette ID-nummeret blir lagret i en cookie for å kunne hentes ut igjen når man går videre til neste trinn. Når man henter ut dette ID-nummeret sørger man for at all informasjon som registreres i de forskjellige trinnene blir knyttet til samme post i databasen.

Et annet sted vi har brukt cookies er ved sjekk av korrekt passord når administrator er inne på de forskjellige administratorsidene. Ved innlogging opprettes en cookie med det krypterte passordet. Denne cookien sjekkes mot kryptert passord i databasen på hver side i administratordelen.

4.5.1 Kode eksempel

Opprettelsen av cookie, dette skjer i trinn2 av registreringen:

```

//Oppretter cookie med navn ID
HttpCookie cookieId = new HttpCookie("id");
Response.Cookies.Clear();

```

```

Response.Cookies.Add(cookieId);

//Legger til den unike ID fra registrering
cookieId.Values.Add(null, idHilsen);

//Levetid er 1 dag.
cookieId.Expires = DateTime.Now.AddDays(1d);

Sletting av cookie, skjer i status.aspx:
//Sletter cookie hvis den finnes
if (Request.Cookies["id"] != null)
{
    // slettes ved å justere dato.
    Request.Cookies["id"].Expires =
    DateTime.Now.AddDays(-1d);

    Response.Cookies.Add(Request.Cookies["id"]);
}

```

4.6 Hjelp

For å hjelpe brukeren gjennom registreringen har vi laget en trinnanvisning i menyen til venstre på hver side i registreringen. Dette gjør at brukeren til enhver tid kan se hvor man er i registreringsprosessen og hva som gjenstår.

Vi valgte også å implementere hjelp ved siden av hvert punkt i form av et spørsmålstegn man kan klikke på. Man får da en beskrivelse av informasjonen man skal fylle inn i gitt felt, samt feltets begrensninger (eks. maks 150 tegn i hilsning).

I tilfelle brukeren skriver inn feil eller ugyldig data i feltene bruker vi forskjellige validatorer på feltene for å sjekke inntastet data. Hvis inntastet data ikke er godkjent får bruker feilmelding om dette. For å forenkle innfyllingen av feltet for hendelsesdato har vi lagt til muligheten for å velge dato fra en kalender. Kalenderen startes av et javascript som eksekveres ved at man trykker på kalenderikonet ved siden av datofeltet.



Figur 20: Hjelpeikon

4.7 Bildehåndtering

Hver hilsning skal bestå av tekst og 2 bilder. Motivet i de to bildene vil være det samme, men det blir lagret i to forskjellige størrelser. En stor versjon som blir brukt i visning av selve hilsningen, og en mindre, kalt thumbnail, som brukes i visning av iframe og visning av kategori.

Vi har valgt å lagre bildefilene direkte på server, mens vi lagrer bildenes filadresse i databasen. Tidlig i planleggingen diskuterte vi

muligheten til også å lagre bilde i databasen, i form av en såkalt blob. Dette ble valgt bort i samråd med veileder da vi ikke så noen åpenbar grunn til å lagre selve bildet i databasen. Når vi lagrer bildene på server som en separat fil vil vi gi direkte aksess til bildefilen. Samtidig vil man ha bildefilene inntakt i tilfelle det skulle skje noe uforutsett i databasen.

Når man velger å laste opp eget bilde sjekkes valgt fil om den har JPEG bildeformat og om filstørrelsen er under 2 megabytes. Hvis filen ikke oppfyller kravene får brukeren melding om dette, ellers kalles funksjonen LastOppFil i Bilde.cs som tar inn den valgte filen. Bilder som blir lastet opp kan selvfølgelig variere i størrelse, derfor har vi satt standard størrelser på de versjonene av bildene som blir lagret. Stort bilde vil ha en maksimal størrelse på 400 x 400 piksler mens thumbnail maksimalt kan være 140 piksler bred, uten begrensning i høyde. Vi har ikke sett det nødvendig å legge inn noen begrensning i høyden da vi bare er avhengig av å kunne passe den inn i iframens satte størrelse, og at det ikke er sannsynlig at noen tar i bruk bilder som er unormalt høye i forhold til bredden. Skulle dette likevel skje vil disse hilsningene bli luket ut i administratordelen.

LastOppBilde funksjonen bruker forholdet mellom den største siden i originalbildet og den største mulige kantstørrelse i stort bilde for å finne en forholds faktor. Først finner funksjonen ut om bildet er størst horisontalt eller vertikalt. Så blir den største siden i det store bildet satt til 400 piksler, mens den minste blir regnet ut som originalbildets minste side multiplisert med forholds faktor. Slik blir forholdet mellom sidene i bildet alltid ivaretatt, og man får ingen strekking av bildet. Samme metodikk blir brukt for å finne høyden i thumbnail bildet, men i thumbnail vil bredden alltid være satt til 140 piksler fordi den skal passe inn i iframen.

For å beholde best mulig kvalitet på bildene ved skalering blir bildene lagt inn i et graphics objekt som man kan sette interpolasjon på. Bildene blir så tegnet opp med ny størrelse og interpolasjon før de lagres på server. Filadressene til både stort bilde og thumbnail skrives så til database. Under ser vi eksempel på skalerte bilder. Først med og så uten interpolasjon.



Figur 21: Med interpolasjon.



Figur 22: Uten interpolasjon.

Til slutt endres bildets oppløsning (dpi) for det store bildet, som senere sendes til avisen. Avisen må ha en viss kvalitet på bildet for at resultatet skal bli tilfredsstillende når de trykker det i avisen. Vanlige foto har oppløsning på 72 dpi. For at bildet skal se bra ut i avisen må denne justeres til 172 dpi.

Hvis ikke brukeren har eller vil laste opp et eget bilde kan man velge å bruke bilder fra galleriet. Man blir da sendt videre til siden kalt galleri.aspx. Her får man presentert de bildene administrator har lagt inn. Man velger et bilde, trykker "Velg merket bilde"-knappen og galleribildets og dets thumbnails filadresse skrives inn i databasen under den gjeldende hilsningen.

4.8 Payex

4.8.1 Generelt

Payex er en elektronisk lommebok som benyttes av mer enn 250 av landets ledende brukersteder. Vi kan nevne eksempler som VG, Dagbladet, TV2, NRK og selvfølgelig Oppland Arbeiderblad. Payex er gratis å opprette og enkel å bruke for brukerne. Payex kan tilby en rekke betalingsmuligheter.

Payex Konto

Payex konto er en elektronisk lommebok for handel på internett. Lommeboken fungerer ved at en bruker fyller den opp med penger via sin bank, nettbank, kredittkort eller Payex Kontantkort. Man kan fylle opp til 1000 NOK på en payexkonto.

Kredittkort

Gjennom Payex betalingsportal får man tilgang til direktekjøp fra VISA og MasterCard. Brukerstedet trenger ikke inngå en egen avtale med kortselskapene, da Payex har en multimerchant avtale alle deres brukersteder benytter. For å kunne handle med VISA/MasterCard trenger brukeren kun kortnummer, utløpsdato og CVC kode.

SMS (betaling med telefonregning)

Payex CPA er en løsning der sluttbruker blir belastet fra 1-60 kroner på sin mobilfaktura/kontantkort ved å få tilsendt en overtaksert SMS. Operatørene krever at man kun benytter dette betalingsmiddelet for kjøp av digitalt innhold.

Sluttbruker trenger ikke å registrere seg på Payex for å handle, men det kreves at de har et norsk mobiltelefonabonnement.

Payex Kontantkort

Dette er en ny løsning for påfylling av Payex kontoen som ble lansert høsten 2004. Man kan kjøpe et elektronisk kontantkort hos Narvesen og deretter fylle på sin konto fra sin mobiltelefon, via internett, på www.payex.no eller via brukerstedene.

Dette vil være spesielt attraktivt for brukersteder med brukere på under 18 år.

Kreditering av transaksjon.

Hvis bruker har betalt en tjeneste som ikke kan leveres, noe som kan være aktuelt i vårt tilfelle, kan bruker få pengene tilbake. Kreditering skjer i Payex sin "Payex Merchant View". Her kan administrator finne frem til spesifikke transaksjoner ved hjelp av transaksjonsnummer.

4.8.2 Payex i vårt system

OA har valgt å la brukerne betale med SMS (telefonregning) eller Payex Konto. Kredittkortbetaling ble også vurdert, men vil ikke bli benyttet da dette medførte en ekstra kostnad på ca.500 kr per måned, pluss transaksjonsgebyr for hver enkelt betaling.

Vårt system skal sende brukeren videre til payex sine sider. Først må vi klargjøre en ordre, et produkt vi vil selge, før vi sender brukeren videre. Når en ordre er generert sendes informasjon om denne til Payex ved å benytte SOAP/Web services. All kommunikasjon med Payex er forøvrig kryptert med en MD5 hash algoritme. Vi trenger også en XML-parser for å oversette informasjonen som Payex returnerer fra de forskjellige metodene. Til slutt bør det foreligge en sjekk på om betalingen har gått bra eller ikke. Dette må gjøres på våre sider. Oppland Arbeiderblad har allerede en konto hos Payex så vi trengte ikke opprette noen ny konto i forbindelse med vårt prosjekt.

All sikkerhet med hensyn til betalingen tar Payex seg av. Payex støtter blant annet 3D-secure. Dette er en sikkerhetsmodell som benyttes av de fleste banker. Denne støtter både Visa og MasterCard. 3D-secure betyr at brukeren blir sendt til respektive banker for yterligere autentisering.

4.8.3 Implementering

For å få aksess til Payex sin Web Service har vi generert en proxy-klasse. Visual Studio .NET har egne verktøy for dette, og ble gjort på følgende måte i et kommandovindu:

```
WSDL http://test.payex.com/pxorder/pxorder.asmx
```

En klasse-fil blir da generert. Denne inneholder alle de nødvendige funksjonene for å opprette og behandle en transaksjon. Funksjonene vi bruker til dette er PxOrder.Initialize() og PxOrder.Complete().

All kommunikasjon måtte i første omgang foregå mot en test server (test.payex.com). Dette er til fordel for begge parter, da ingen faktisk betaling blir utført og alle feil kan lukes vekk.

All kommunikasjon med Payex er kryptert med en MD5 hash algoritme. Dette blir gjort ved hjelp av ferdige klasser i .Net.

Krypteringsnøkkelen ble laget ved opprettelse av test-miljøet hos Payex.

Vi kunne velge mellom to forskjellige transaksjoner, 1-fase og 2-fase. Vi valgte 1-fase der brukeren blir belastet med en gang. Dette ble definert i `PxOrder.Initialize()` ved å sette attributtet `PurchaseOperation = SALE`.

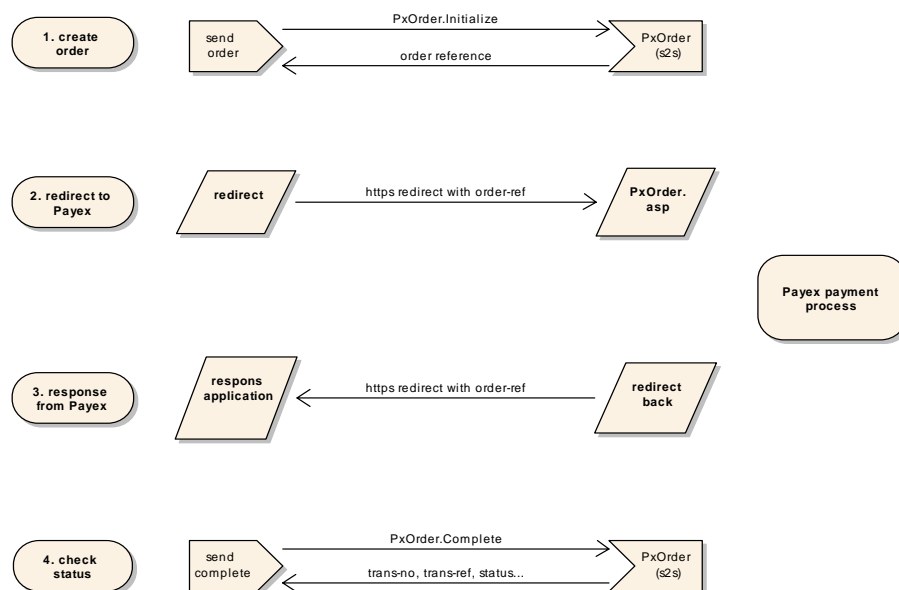
Det var to forskjellige måter å implementere kjøpsprosessen på. En såkalt re-direct modell og en direct modell. I vårt tilfelle valgte vi å benytte oss av re-direct modellen. Dette betyr at brukeren blir sendt fra OA sine sider til Payex sine sider for utføring av selve betalingen. Når betaling er utført blir brukeren sendt tilbake til OA. I direct modellen, som vi ikke valgte, blir brukeren på OA sine sider under hele betalingsprosessen.

Ved bruk av re-direct modellen vil en ordre bestå av kontonummer, ordrenummer, pris, ID og produktbeskrivelse. Denne informasjonen samles i en datapakke sammen med en såkalt hash-nøkkel. Til slutt blir samtlige data kryptert ved hjelp av en MD5-algoritme. Krypterte data sendes til `PxOrder.Initialize()`. Denne funksjonen returnerer en XML-tekst med informasjon om operasjonens utfall. Vi kjører XML-teksten gjennom en XML-parser for å hente ut status for initiering, eventuelle feilmeldinger og referansenummer for orden. Dette nummeret sender vi som en querystring til Payex sine sider, der transaksjonen vil bli utført (`test.payex.com/Order.asp?=&OrderRef`). Når betalingen er gjennomført sendes brukeren tilbake til vår statusside. Her blir betalingen sjekket ved hjelp av `PxOrder.Complete()`. Vi krypterer en ny datapakke, denne gangen med kontonummer og ordreferanse på samme måte som under initieringen. Vi får på nytt returnert en XML-tekst med informasjon om betalingen. Er betalingen godkjent vil hilsningen flyttes bort fra den temporære databasen. Det eneste som nå gjenstår før hilsningen er på nett, er at administrator godkjenner den.

For å forhindre misbruk av den returnerte URL, vil ingen informasjon om transaksjonen ligge i denne. Dette er grunnen til at det er viktig å gjøre en sjekk på transaksjonen ved hjelp av `PxOrder.Complete()`.

Når bruker dirigeres fra Payex sin server til vår statusside vil vedkommende få en melding om at man forlater et sikkert område. Trykker bruker "Nei" her, vil vår løsning aldri motta informasjon om at transaksjonen er gjennomført, noe som betyr at hilsningen aldri vil bli sendt til godkjenning. Grunnen til at vi får denne meldingen er at Payex kjører sine tjenester på en sikker server, HTTPS. Måten vi valgte å løse dette på ble å kjøpe et SSL-sertifikat for Oppland Arbeiderblads server. Ved å bruke SSL på statusside vil vi hindre denne meldingen, siden vårt område da også vil være sikkert (HTTPS).

4.8.4 Kommunikasjonen mot Payex



Figur 23: Figur fra Payex sin implementasjonsmanual.

Etter å ha valgt hvilke tjeneste som ønskes skjer følgende i kommunikasjonen mot Payex. Figuren over viser dette på en forklarende måte.

1. Klikk "Betal". Linken går til et lokalt server-script som genererer en kryptert betalingsforespørsel.
2. Brukeren sendes til Payex med den krypterte forespørselen.
3. Login / Betal siden hos Payex vises.
4. Payex genererer en kryptert betalingskvittering som sendes med brukeren tilbake til vår bekreftesside.
5. Vår applikasjon dekrypterer og validerer.

4.8.5 SSL

SSL (Secure Sockets Layer) er koding av data som sendes over nettet. Hvis du kjøper varer på nettet er det viktig at sensitiv informasjon som f.eks. kredittkortopplysninger blir utilgjengelig for uvedkommende. Hvis nettbutikken du handler i benytter SSL standarden, vil den informasjonen du legger igjen bli kodet når informasjonen går over nettet.

I vår løsning vil bruk av SSL være relevant fordi Payex benytter seg av dette. Når bruker da dirigeres tilbake fra Payex til vår side for å få bekreftet transaksjonen, vil man få en sikkerhetsadvarsel med spørsmål om bruker vil forlate det sikre området. Klikker brukeren "Nei" her, vil hilsningen aldri bli registrert som godkjent og den vil ikke bli sendt til godkjenning. Installerer vi derimot SSL og bruker dette på vår bekreftesside, vil man slippe denne advarselen og vil utelate risikoen for at bruker ikke får godkjent hilsningen.

4.8.6 Kodeeksempler

Eksempel på xml-retur fra PxOrder.Initialize():

```
<?xml 20version="1.0" 20encoding="utf-8" 20?>
<payex>
  <header 20name="Payex 20Header 0v1.0">
    <id>3f94163017c0465787475717d7a89fdc</id>
    <date>2005-03-17 2015:09:18</date>
  </header>
  <status>
    <code>OK</code>
    <description>OK</description>
  </status>
  <orderRef>a9c84b91c50f49159f9356193e4ad4a6</orderRef>
  <sessionRef>e0f6c10313f24201af42d751d105dc39
  </sessionRef>
</payex>
```

MD5Hash.cs

```
using System.Security.Cryptography;

public bool Hash(string data, ref string hash)
{
    try
    {
        //Ny "MD5-haser" opprettes og initieres.
        MD5CryptoServiceProvider hasher = new
        MD5CryptoServiceProvider();
        hasher.Initialize();

        //Det som skal "hashes" lagres i en byte
        //array.
        byte[] bytes;
        bytes =
        hasher.ComputeHash(Encoding.Default.GetBytes
        (data));

        StringBuilder sb = new StringBuilder();

        for (int i=0;i<bytes.Length;i++)
        {
            sb.Append(bytes[i].ToString("X2"));
        }
        hash = sb.ToString();
        return true;
    }
    catch (Exception e)
    {
        return false;
    }
}
```

XMLParser.cs

```
using System.Xml;

//Henter ut en bestemt XML node fra XML-tre.
public string ParseRes(string xmlText, string node)
{
    string nodeRes = "";
    XmlDocument doc = new XmlDocument();
    doc.LoadXml(xmlText);
    XmlNode myNode = doc.SelectSingleNode(node);
    nodeRes = myNode.InnerText.ToString();
    return nodeRes;
}
```

4.9 Saxotech

Vår oppgave er å sende hilsning med bilde og hilsningstekst inn til avisens nettdesksystem, Saxotech. Her sitter en av avisens designere og redigerer hilsninger inn i nyeste utgave av avisen.

4.9.1 Hva er Saxotech?

Saxotech er en global utgiver av programvare for aviser og andre publikasjoner. Saxotechs programvare hjelper publikasjoner med måten de lager, styrer og publiserer innhold. Hovedmålet er å redusere kompleksiteten i publikasjonsprosessen. Programvaren brukes av kjente aviser som Washington Times, Dagens Næringsliv m.fl.

4.9.2 Implementering

Når administrator godkjenner en hilsen som skal til avisens papiirutgave må systemet vårt ta seg av dette. Det blir da sendt e-post med bilde som vedlegg til Saxotech. For sending av e-post brukte vi standardklasser i Visual Studio .NET. (system.web.mail).

Hos OA havner e-posten i en bestemt mappe i Saxotech. Først henter vi ut all informasjon om hilsningen som skal sendes før vi henter Saxotechs e-postadresse fra databasen.

Hilsningsteksten skrives til bildets metadata, såkalt EXIF-informasjon. Hilsningsteksten blir lagret i bildets captionfelt før e-posten sendes. Det var først meningen at vi skulle sende hilsningen ved hjelp av en e-post med spesielle koder, men dette viste seg å ikke fungere som det skulle i Saxotech. Løsningen ble derfor å benytte seg av bildets EXIF-felt. Dette er samme måten som avisens freelancere benytter seg av. De benytter Photoshop eller et liknende program der de legger artikkelens tekst til bildet. Dette er en felles standardmåte å gjøre det på innad i bransjen.

Når bildet så kommer til Saxotech er det enkelt for designeren som sitter og lager avisen å implementere hilsningen. Vi hadde litt problemer med å få frem hilsningsteksten sammen med bildet, men når bildet ble satt inn i Quark (avisens tekst/layoutprogram) viste det seg at dette kom automatisk.

4.9.3 EXIF

EXIF står for *Exchangeable image file format* og er en standard for lagring av tekstinformasjon i bildefiler. De fleste av dagens digitalkameraer lagrer informasjon om bilde og kameratype etc. Eksempler på dette kan være alt fra bildebeskrivelse til dpi, dato og klokkeslett. Bildetyper som støtter denne typen informasjon er blant annet JPEG og TIFF.

Alle felt i EXIF-delen av bildet har sine egne bestemte hexadesimale identifikasjonsnummere.

Eksempler på forskjellige IDer i EXIF:

0x010e - bildebeskrivelse
 0x0132 - dato/klokkeslett
 0x8298 - copyright

4.9.4 Kodeeksempler

```
//Funksjoner som skriver EXIF-informasjon:
private static ImageCodecInfo HentEncoderInfo(String
    mimeType)
{
    //Returnerer informasjon om Encoder brukt i bildet.
    int i;
    ImageCodecInfo[] encoder;
    encoder = ImageCodecInfo.GetImageEncoders();
    for(i = 0; i < encoder.Length; ++i)
    {
        if(encoder[i].MimeType == mimeType)
            return encoder[i];
    }
    return null;
}

public static void SkrivNyBeskrivelseTilBilde(string
    filnavn, string nyBeskrivelse)
{
    System.Drawing.Image bilde;
    PropertyItem[] PropertyItems;

    //Lagrer den nye beskrivelsen i en byte array.
    byte[] bBeskrivelse = new
        Byte[nyBeskrivelse.Length];

    string filnavnTemp;

    //Encoder informasjon.
    System.Drawing.Imaging.Encoder Enc =
        System.Drawing.Imaging.Encoder.Transformation;

    EncoderParameters EncParms = new
        EncoderParameters(1);

    EncoderParameter EncParm;
    ImageCodecInfo CodecInfo =
        HentEncoderInfo("image/jpeg");

    //Kopierer beskrivelse til en byte array.
    int i;
    for (i=0;i<nyBeskrivelse.Length;i++)
        bBeskrivelse[i] = (byte)nyBeskrivelse[i];

    //Henter bildet som beskrivelsen skal skrives til.
    bilde = System.Drawing.Image.FromFile(filnavn);

    //Skriver bildebeskrivelse til riktig "property"
    PropertyItems = bilde.PropertyItems;
    PropertyItems[0].Id = 0x010e;
```

```

//Caption, 0x010e hentet fra EXIF standard.
PropertyItems[0].Type = 2;
PropertyItems[0].Len = nyBeskrivelse.Length;
PropertyItems[0].Value = bBeskrivelse;
bilde.SetPropertyItem(PropertyItems[0]);

//Kan ikke lagre i samme bildet, må bruke et
temporært bilde i stedet.
filnavnTemp = filnavn+".temp";

//For at bildet ikke skal miste kvalitet, er det
viktig å rotere 90grader.
EncParm = new EncoderParameter
    (Enc, (long)EncoderValue.TransformRotate90);
EncParms.Param[0] = EncParm;

//Skriver den nye beskrivelsen til det roterte
bildet.
bilde.Save(filnavnTemp, CodecInfo, EncParms);

//Frigjør allokert minne.
bilde.Dispose();
bilde = null;
GC.Collect();

//Sletter originalbilde, men vil bli erstattet
senere.
System.IO.File.Delete(filnavn);

//Rotere tilbake bilde og lagrer.
bilde=System.Drawing.Image.FromFile(filnavnTemp);
EncParm = new EncoderParameter
    (Enc, (long)EncoderValue.TransformRotate270);
EncParms.Param[0] = EncParm;
bilde.Save(filnavn, CodecInfo, EncParms);

//Frigjør minne.
bilde.Dispose();
bilde = null;
GC.Collect();

//Sletter det temporære bilde.
System.IO.File.Delete(filnavnTemp);
}

```

4.10 Escenic

Escenic er et styringsprogram for innhold på websider. Programmet inneholder muligheter for generering av avanserte internettsider. Escenic brukes av kjente bedrifter og organisasjoner som Filmweb, Norgesgruppen, TV2 Norway m.fl.

Den eneste tilknytningen vårt prosjekt har mot Escenic er vår iframe som skal vises på nettavisens forside. Vårt ansvar er at denne har korrekt innhold, ansvar for visning og linking mot iframe ligger hos OA.

5 Testing og kvalitetssikring

5.1 Kompatibilitet

Oppdragsgiver har ikke stilt spesifikke krav til kompatibilitet i løsningen. Vi har utviklet og testet applikasjonen i et Microsoft-miljø. Applikasjonen skal fungere optimalt i nyeste versjon av Microsoft Internet Explorer.

5.2 Kvalitetssikring

5.2.1 Sikring av data og backup

Vi har gjennom hele prosjektet vært klar over faren ved tap av data og tatt hensyn til dette. Ingenting er mer demotiverende enn å miste flere ukers arbeid. Det ble derfor tidlig vedtatt at en person i gruppa hadde ansvar for å ta backup av alle data to ganger i uken. Nyeste backup har hele tiden vært tilgjengelig for resten av gruppa på prosjektets hjemmeområde hos HiG.

Vi vurderte bruk av konfigurasjonsstyringsverktøyet CVS i begynnelsen av prosjektet, men prioriterte ikke å bruke tid på dette da vi gjennom hele prosjektet har jobbet sammen på grupperom. Synkronisering av kode medførte ikke nevneverdig ekstraarbeid uten CVS. Totalt sett ville vi antakeligvis brukt like mye tid på å sette oss inn CVS som vi har brukt ekstra tid på å synkronisere og oppdatere kode.

5.3 Testing

5.3.1 Testing av kildekode

Arbeidet med koding, design og testing har foregått relativt parallelt gjennom iterasjonene. For hver endring og tillegg i funksjonalitet har vi testet om det har gitt ønsket effekt. Dette er testing som gruppemedlemmene har tatt seg av.

Foran møtene med OA har vi lagt ut nyeste fungerende versjon av applikasjonen sånn at oppdragsgiver har kunnet teste denne. På denne måten har også oppdragsgiver vært involvert i testing av applikasjonen. Gjennom dette har vi fått tilbakemelding på om produktet møter satte krav og ønsket funksjonalitet.

5.3.2 Testing av bruker- og administrasjonsdel

Funksjonaliteten i bruker- og administrasjonsdel er stort sett den samme. Stort sett alt baserer seg på kommunikasjon mot database. Vi har testet hvert enkelt element i applikasjonen etter hvert som disse er implementert.

I brukerdelen har vi lagt spesielt vekt på at det skal være umulig å få applikasjonen til å krasje. Uansett hva man fyller inn og hvor innstilt man er på å ødelegge applikasjonen, skal den veilede brukeren inn på rett spor igjen med forklarende og konstruktive feilmeldinger.

5.3.3 Testing mot Payex

Selv utfører vi generell testing på funksjonaliteten, men hovedtestingen vil Payex ta seg av, de skal sikre at alt er som det skal før de godkjenner applikasjonen. Dette gjør de gjennom en såkalt akseptansetest. I denne testen var det visse punkter som måtte være til stedet for å få applikasjonen godkjent. Dette innebefattet blant annet at vi måtte ha Payex sin logo på siste side, før bruker ble sendt til betaling. På denne siden måtte vi også ha diverse linker for opprettelse av konto og generell informasjon om Payex.

5.3.4 Testing mot Saxotech

Når det gjelder testingen av hvordan systemet vårt fungerer med Saxotech, blir det testet av designere som sitter i redaksjonen hos OA. Vi har fått opprettet en testmappe i systemet der vi kan sende inn våre hilsninger via e-post. Denne testmappen fikk vi innblikk i under ett av våre besøk hos OA.

6 Installasjon og realisering

6.1 Introduksjon

Det har vært et mål for oss å gjøre løsningen så funksjonell og intuitiv som mulig. Dette for at alle med en grunnleggende datakunnskap kan bruke det og dermed kjøpe en hilsning som blir liggende på nett eller i avisens papirutgave. Dessuten har det også vært viktig at administrator enkelt kan gjøre vedlikehold og drifte løsningen.

6.2 Installasjon

Vi fant ut at det var hensiktsmessig å lage en installasjonsfil for løsningen. Denne filen inneholder alle filer og katalogstruktur som er nødvendig for å kjøre løsningen. Denne filen må først kjøres lokalt på en maskin før hele katalogstrukturen må kopieres manuelt ut på server ved hjelp av en FTP-klient. Installasjonsfilen ligger vedlagt på CD-platen.

Når filene er lagt ut på server må man legge inn en rekke innstillinger gjennom vår installasjonsside. Her settes SQL-, avis-, server- og Payex informasjon. Forutsetningene for å bruke løsningen vår er at man må ha en web-server som støtter ASP.NET og MySQL.

For en grundigere beskrivelse av installasjonsprosessen, se brukermanualen i vedlegg E.

6.3 Vedlikehold

Når det gjelder vedlikeholdet av løsningen vår gjøres dette stort sett gjennom administratordelen. Her kan administrator slette og endre hilsningene. Man kan også opprette kategorier og oppdatere galleriet. På denne måten holder man liv i løsningen. Vil man fjerne løsningen fra webserveren er det bare å slette katalogen man la inn ved opprettelse av løsningen.

7 Beskrivelse av utviklingsprosessen

7.1 Om inkrementell systemutvikling i prosjektet

7.1.1 Inkremitter:

Inkrement	Innhold
Lage nettsted	Planleggingen og konstruksjonen av rammeverket systemets websider ble bygget på.
Kommunikasjon med database	Opprette kommunikasjon til og fra database, gjennom systemets websider.
Kommunikasjon med Saxotech og Payex	Sende hilsninger på e-post til papirutgaven av avisen. Kommunikasjon med betalingstjenesen Payex.
Administrasjonsmodul	Konstruksjon av administrator sidene med innstillingsmuligheter og kontroll over systemet.
Ferdigstilling	Testing av hele systemet, utbedringer. Rapportskrivning.

Tabell 2: Inkremitter

Inkrementene ble utført fortløpende som beskrevet i listen over, dog med avvik noen steder. Rekkefølgen på inkrementene ble satt opp etter den logiske grunn at inkrementene bygger på hverandre. Utviklingen har hatt fokus på funksjonalitet og ikke design, derfor er designinkrementet plassert sent i utviklingen. Det ble satt av lite tid til dette.

7.2 Utviklingen av inkremitter

Hvert inkrement startet med at vi hadde en gjennomgang av kravene knyttet til inkrementets innhold. I denne fasen av inkrementet ble implementeringsløpet lagt sammen med strategier for løsninger. Her kunne det hende at vi kom bort i problemer som var uklare i kravspesifikasjon, og trengte utbrodering. I slike situasjoner kontaktet vi enten oppdragsgiver eller veileder for å diskutere problematikken. I enkelte situasjoner tok vi også slike avgjørelser selv. Oppdragsgiver gav oss tidlig frie tøyler til å ta avgjørelser på egenhånd, og sa at det heller ville komme innspill fra deres kant hvis det var noe som måtte endres.

Det har også forekommet at ønsker om ny funksjonalitet har dukket opp i løpet av inkremitter. Dette har kommet både fra prosjektgruppens, veileders og oppdragsgivers side. Eksempelvis mulighetene for å legge til og fjerne kategorier og galleriet med utskiftbare bilder er ønsker vi har implementert som et resultat av at man har sett flere muligheter etter hvert, og at det har blitt et felles ønske fra flere parter.

I og med at mesteparten av utviklingen vi har gjort har vært stoff vi ikke har hatt kjennskap til fra før, har det meste krevd en del forskning. Mye tid har vært brukt på prøving og feiling for å oppnå inkrementers ønskede effekt. Vi har gjort god bruk av Microsofts dokumentasjon av .NET plattformen, diverse bøker og også søkt hjelp via internett i form av eksempler på fremgangsmåter og dokumentasjon.

Etter hvert som elementene har falt på plass har vi testet fortløpende, og foretatt de justeringene som trengs. Hvert inkrement har blitt startet og avsluttet med at vi har dokumentert innholdet i form av rapportskrivning. Utvikling av rapport underveis har hele tiden vært en del av prosjektplanen, noe som gjør at man unngår en voldsomt travel periode i slutten av prosjektet hvor man må dokumentere alt i løpet av altfor kort tid.

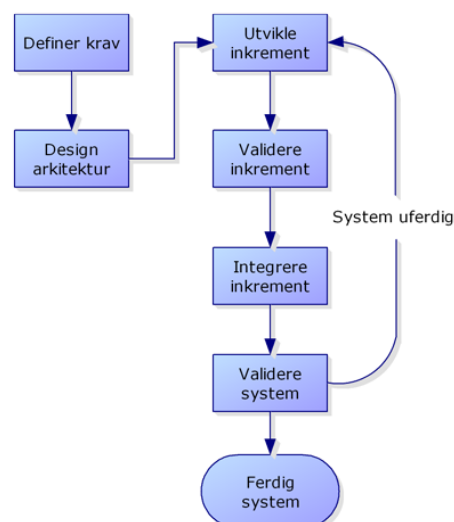
7.3 Validering og integrering av inkremitter

Validering av de forskjellige inkrementene har blitt foretatt når de var ferdig integrert og testet sammen med resten av applikasjonen. Her har fokus vært på å avdekke feil og mangler både innad i inkrementet, og i dets interaksjon med tidligere inkremitter. I denne prosessen av inkrementene har det til tider vært nødvendig med småjusteringer. Dette gjorde seg kanskje spesielt gjeldende ved skriving og lesing til og fra database. En ferdig validering av systemet ble utført ved slutten av utviklingen for å verifisere at alle inkremitter fungerte sammen.

7.4 Vår anvendelse av inkrementell systemutvikling

Gjennom utviklingen har vi fordelt oppgavene innad i inkrementene slik at hver person har hatt sitt fokuspunkt. I en så liten gruppe har dette fungert veldig bra, og dette har gjort det enklere å høste resultater enn hvis alle skulle ha fullt innsyn i all utvikling.

Inndelingen av inkrementene har fungert bra. Dette mener vi kommer av at vi har basert oss på en enkel og logisk fordeling hvor målene og ønsket effekt av hvert inkrement er veldig klar. Samtidig må det nevnes at det har vært nødvendig å starte enkelte inkremerter før foregående inkremerter har vært ferdigstilt. Dette gjelder blant annet inkrementet for kommunikasjon med database som ble delvis implementert mens vi laget websidene, da dette ble en naturlig progresjon av prosjektet. Noe av grunnen til dette er at en del oppgaver kunne være vanskelig å løse og ble utsatt, eller at det var naturlig at deler av gruppen gikk videre til neste inkrement mens andre fullførte påbegynt inkrement. For eksempel bildehandteringen, som var en del av kommunikasjon med database inkrementet, tok litt lengre tid en planlagt og strekte seg over et større tidsrom en vi hadde sett for oss forut for iterasjonen. Dette var mest på grunn av prosjektgruppens ønske om å bruke interpolasjon på bildene i systemet. Å gå videre til nye inkremerter uten at tidligere inkremerter er ferdigstilt kan sees som en risikabel strategi som kan føre til at man kan sitte igjen med en del kode og oppgaver som er halvferdig når man går mot prosjektets slutt. Dette ble ikke tilfellet for oss da de oppgavene som ble utsatt var ansett som relativt små, men var vanskelige å løse i øyeblikket. Og det viste seg å være en fordel å kunne legge slike problemer fra seg og komme tilbake til de på et senere tidspunkt. Når det hadde gått litt tid kunne man gjerne se problemet med litt andre øyne og man hadde gjerne også litt mer kunnskap om koding generelt. Det var også greit å kunne involvere andre gruppemedlemmer og eventuelt veileder for å kunne få synspunkt på problemer og også bidrag til nytenkning som hjalp til progresjon.



Figur 24: Progresjon innad i inkrementene.

Selv om noen av inkrementene har blitt lengre enn planlagt har vi fulgt det vanlige progresjonen innad i inkrementer som illustrert i figuren.

8 Diskusjon av resultater

8.1 Resultater

Vi mener vi har fått til en bra løsning og er stolte av applikasjonen vår. Det finnes selvfølgelig problemer vi gjerne skulle ha løst på andre måter. Men her bør man ta i betraktning at vi har vært nødt til å tilegne oss mye kunnskap underveis i prosessen.

Et bevis på at applikasjonen er positivt mottatt hos Oppland Arbeiderblad er at de i skrivende øyeblikk har denne inne for godkjenning. Vår kontaktperson har meddelt oss at applikasjonen kanskje er i bruk av OA ved prosjektets avslutningsdato.

8.1.1 Hva ble bra

- **Registrering:** Vi har laget en oversiktlig og sikker måte å registrere hilsningene på. Vil for eksempel brukeren ha hjelp, finner man dette ved å klikke på spørsmålsteget.
- **Payex:** En meget god og oversiktlig måte for betaling over internett. Sikkert og pålitelig.
- **Vedlikehold:** Det er enkelt å holde løsningen i "live", ved å legge til, slette eller endre kategorier. Det samme gjelder galleribildene.
- **Bildekvalitet:** De skalerte bildene blir veldig bra etter interpolasjonen. I tillegg tar bildefilene veldig liten plass. Den minste versjonen er i snitt 3kb stor, mens den store er oppunder 20kb. Vi oppnår altså god kvalitet og bruker likevel lite plass på server.
- **Oversiktlig løsning:** Etter å ha fått en del nyttige tilbakemeldinger gjennom vår uformelle spørreundersøkelse føler vi at vår applikasjon er lettere å bruke og mer oversiktlig enn vg.no sin løsning.
- **Oppland Arbeiderblad er fornøyd:** Vår oppdragsgiver har gjennom hele prosessen vært positive til løsningen vår. Applikasjonen skal nå gjennom en godkjenningsprosess i avisen før den publiseres på deres nettsider. Andre aviser under A-Pressen har også vist sin interesse. En løsning som brukes aktivt av en eller flere nettaviser vil være en utmerket referanse for oss.

8.1.2 Hva ble ikke så bra

- Flerbruker: Vi kunne helt klart ha løst flerbruker-problematikken på en annen måte. For eksempel kunne vi brukt en felles server og lagt implementering av nye aviser i administrasjonsmodulen. Grunnen til at vi ikke gjorde dette, var at det tok et par måneder før vi fikk klarsignal på at A-Pressen ville være hovedoppdragsgiver. Dette fremkommer også av oppgavetekst og hovedmål for prosjektet der det presiseres at en ferdig implementert løsning hos oa.no er hovedprioritet. Vi lagde derfor først en applikasjon som fungerte 100% for OA før vi startet å lage flerbrukerløsningen rundt denne. Sånn det er i dag må hele applikasjonen kopieres og tilpasses en ny avis. Dette har vi prøvd å gjøre enklest mulig, blant annet ved hjelp av en forklarende implementasjonsguide/brukermanual. (Vedlegg E).
- Valg muligheter: Brukeren kunne hatt større valg muligheter når det gjelder manipulasjon av bildene. For eksempel kunne det vært mulig å gjøre utsnitt av bilde innad i applikasjonen.
- Serveren må ha MySQL, ASP.NET og SSL-støtte. Noe som medfører ekstra kostnader for avisen. Hadde vi for eksempel kodet i PHP, ville løsningen blitt rimeligere.

8.1.3 Ytelseskrav

Som nevnt tidligere i rapporten (kapittel 2.3.1) hadde ikke oppdragsgiver noen spesielle ytelseskrav utenom at det skal virke raskt og effektivt. Dette er blitt realisert ved å holde alle bildestørrelser nede til et minimum uten at bildekvalitet har blitt svekket, både når det gjelder designet og bildet i hilsningene.

8.1.4 Problemer underveis

I forkant av prosjektet hadde vi satt problemer med implementering av Saxotech som ett risikomoment. Vi gikk derfor tidlig ut og hentet inn dokumentasjon om hvordan dette skulle gjøres. I følge dokumentasjonen var dette veldig greit, noe det ikke viste seg å være i praksis da OA hadde gitt oss feil informasjon. På et møte i April fikk vi på ny beskjed om hvordan Saxotechproblemet skulle løses. Denne gangen ved å editere EXIF-informasjonen i bildene vi sendte og legge inn hilsningen her. Dette fungerte heller ikke. Bildet kom frem, men ikke hilsningsteksten. Først i begynnelsen av mai fikk vi den endelige løsningen på problemet da det viste seg at det var en feil hos OA som gjorde at EXIF-informasjonen ikke ble vist. Dermed slapp vi gjøre ytterligere endringer i koden vår.

Et annet problem vi også brukte en del tid på var interpolasjon på bildene. Her dreide det seg stort sett om å finne rett måte å gjøre dette på.

Totalt sett har vi lært at gjennomgang av kritiske punkter bør gjøres tidligst mulig i prosjektet. Dette gjelder spesielt problemet med Saxotech der vi trodde vi hadde full kontroll, men fikk oss en overraskelse da det ikke fungerte likevel.

8.1.5 Hva virker ikke

Gjennom testing har vi opplevd enkelte småfeil dersom man kjører applikasjonen i andre nettlesere enn Internet Explorer. Dette gjelder i all hovedsak tekst og tekstboksstørrelser som ikke forholder seg slik vi i utgangspunktet hadde laget dem. Det har også vist seg at den enkelte validatorer og javascrpts i Visual Studio ikke fungerer i alle nettlesere. Feilene vi har oppsporet hindrer likevel ikke brukeren i å legge inn hilsen og bruke applikasjonen på en normal måte.

8.2 Alternative løsninger

En alternativ løsning kunne ha vært og utviklet løsningen på et annet språk, som for eksempel PHP. Innad i MS Visual Studio kan man også velge andre språk enn C#, nemlig C++, J++ og Basic. Grunnen til at vi valgte Visual Studio var at vi tidligere hadde noe erfaring med dette verktøyet. Språkvalget falt på C# fordi dette virket som en god utfordring og grei ballast å ta med seg videre. I tillegg har vi også brukt en del javascrpts.

8.2.1 Forslag til forbedringer og nye oppgaver

Noe som kan forbedres ytterligere er måten vi har taklet bruken av flerbrukere. Det ligger stort plassbesparende potensial i å lage alt i en felles applikasjon, og ikke kopiere hele løsningen for hver avis som eventuelt skal benytte den.

Ved opplasting av bildene kunne man laget et verktøy for manipulasjon. Her kunne man lagt til farge, rammer, tekster og figurer. Vi kunne også lagt inn funksjonalitet der brukerne kunne kommentert hilsninger og sendt SMS-tips til venner og bekjente.

8.3 Evaluering av gruppens arbeid

8.3.1 Innledning

Alle gruppemedlemmene hadde tidligere flere prosjekter bak seg, og har samarbeidet i mange av disse. Dermed visste vi hva gruppemedlemmene stod for og hva de var gode for. Det ble ikke noe problem å få gruppen til å fungere godt sammen. Gruppen har

vært samlet i store deler av prosjektet, noe som var hensiktsmessig med tanke på kontroll og kommunikasjon innad.

8.3.2 Organisering

Gruppen har hatt følgende delegerte ansvarsområder:

- Prosjektleder: Leder for prosjektet, sørger for å fordele arbeidsoppgaver, og at ting ikke forblir ugjort
- Kontaktperson: Gruppens ansikt utad til veileder, oppdragsgiver og ellers.
- Loggansvarlig: Skriver logger fra alle møter og hvordan dagene har utartet seg.
- Backupansvarlig: Backup av alle dokumenter.

Ansvarsområdene har vært fordelt i samsvar med det som ble planlagt under forprosjektet.

8.3.3 Fordeling av arbeid

Vi har fordelt arbeidsoppgavene slik at den/de som har størst kompetanse innen et felt får jobbe med oppgaver innenfor dette. Dreide det seg om et felt der ingen hadde kompetanse ble den/de med mest lyst og ønsker prioritert.

8.3.4 Tidsforbruk

Vi har gjennom prosjektperioden ført logg over tidsforbruk per person og totalt for gruppen. Totalt timeantall per person er som følger:

Tom André:	506 timer.
Espen:	423 timer.
<u>Atle</u>	<u>474 timer.</u>
<u>Totalt:</u>	<u>1403 timer.</u>

8.3.5 Prosjekt som arbeidsform

Alle gruppemedlemmene hadde som sagt erfaring med prosjekter fra tidligere, men ingen av dem har vært i samme størrelsesorden som dette. Vi har tatt det som en utfordring og føler vi har løst det på en tilfredsstillende måte. Helt knirkefritt har det ikke gått og rom for forbedringer er det helt sikkert. Alt i alt har vi likevel lært veldig mye. Alt i fra hvordan en bør gjennomføre et prosjekt på denne størrelsen til hvordan det fungerer å jobbe i gruppe over lengre tid og bli enige om problemer som dukker opp.

8.3.6 Subjektiv opplevelse av prosjektet

Arbeidet med prosjektet har vært en utfordrende erfaring. Med dette setter vi prikken over i'en etter 3 år på HiG. Kommunikasjonen mellom deltakerne har hele tiden vært løs og

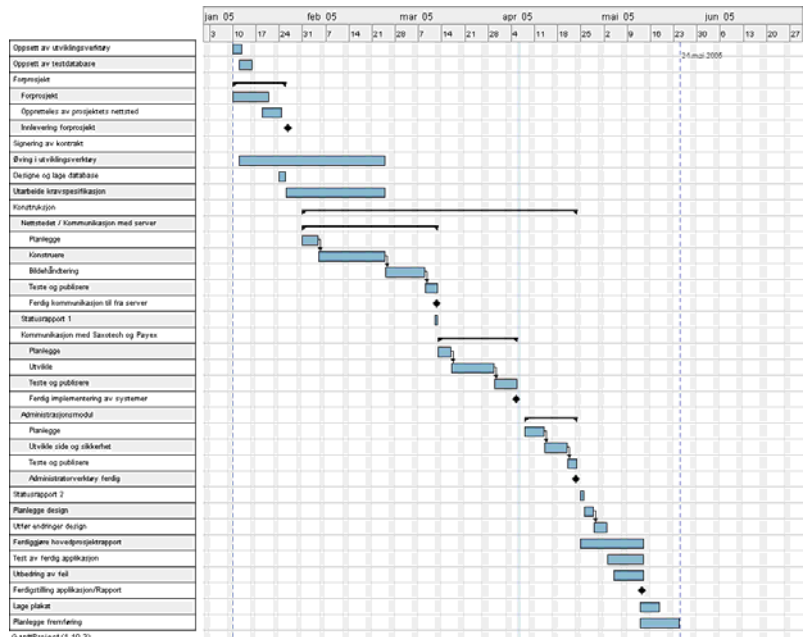
ledig, vi har klart og beholde roen når det til tider har vært hektisk og lange dager har tatt mye av fritiden. Totalt sett har vi vokst på et slikt prosjekt, noe som forhåpentligvis gjør oss mer forberedt på hva som venter i arbeidslivet.

8.4 Drøfting av fremdriftsplan

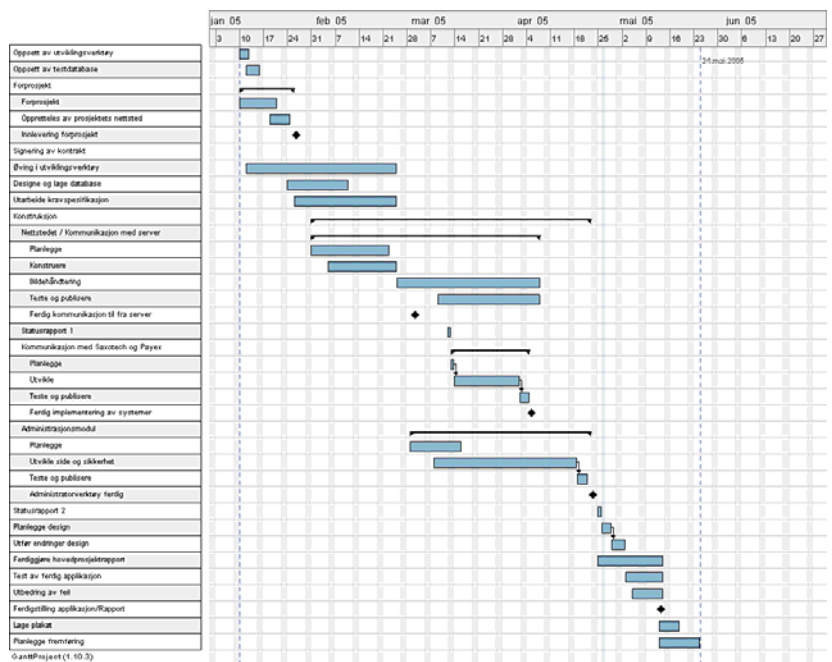
Generelt sett har vi fulgt fremdriftsplanen godt (se figur 25 og 26). De avvikene man ser i før og etter ganttskjemaene kan se ganske grove ut på skjemaet. Men i virkeligheten er dette bare små deler av funksjonaliteten som ikke var ferdig implementert, og aktivitetene har da blitt forskjøvet en del. For eksempel designet av databasen tok litt lengre tid en tidligere planlagt pga. endringer i struktur og behov for tilleggsvariable. Det samme gjelder i første iterasjon der planleggingen av nettstedet og bildehandteringen dro ut en del. Planleggingen, på grunn av litt strukturelle omlegginger og bildehandtering, på grunn av problematikken rundt interpolering.

Den største forskjellen i de to skjemaene finner vi i at administratormodulen ble påbegynt tidligere en planlagt. Allerede sent i februar så vi behovet for å kunne administrere vårt eget system. Vi satte en person på å planlegge å begynne implementeringen av de administrative verktøyene. Samkjøring av denne iterasjonen og utviklingene av kommunikasjon med eksterne systemer gikk veldig bra da oppgavene ble fordelt naturlig innad i prosjektgruppen.

I og med at dette er det første store prosjektet vi har utført, føler vi at vi har truffet ganske godt med fremdriftsplanen. Avvikene kommer som en naturlig følge av mangelen på erfaring med planlegging over en lengre prosjektperiode og mangelen på kunnskap om hvor lang tid aktivitetene ville ta. I innledende fase følte vi at for mye tid ble brukt på å få systemer opp å gå og å lære å bruke utviklingsverktøyene, men dette var kunnskap som man bare måtte erverve seg en gang. Og etter hvert i utviklingen dro man kunnskap med seg inn i nye iterasjoner. Man fikk erfaring med hvordan man skulle gripe an et problem, og produktiviteten gikk betraktelig opp.



Figur 25: Gantt-skjema fra forprosjekt rapport.



Figur 26: Gantt-skjema, revidert utgave etter utviklingsprosess.

9 Konklusjon

Gjennom et halvt års arbeid har vi jobbet med et spennende og variert prosjekt, helt fra oppdragsgivers presentasjon til ferdig implementering av løsningen. Prosjektet satte krav til planlegging, samarbeid og motivasjon i tillegg til faglige kunnskaper. Motivasjonen til gruppens medlemmer har gjennom hele prosjektet vært god. Oppgaven i seg selv har vært motiverende å jobbe med da vi hele tiden har sett resultater av arbeidet vårt og fått positive tilbakemeldinger fra oppdragsgiver. Vi har lært veldig mye om prosjektstyring gjennom dette prosjektet og ser på det som en nyttig erfaring.

Ingen i gruppen har noen gang jobbet med et så stort og komplisert prosjekt. Erfaringen fra å ha angrepet et ukjent prosjekt der vi også har måttet sette oss inn i et nytt kodespråk har vært veldig god. Det å kunne tilegne seg ny kunnskap og bruke denne i et prosjekt er noe vi helt sikkert vil oppleve i arbeidslivet. Vi har også fått erfaring i å kommunisere med forskjellige aktører. Dette gjelder i hovedsak Payex og Webhuset.

Vi sitter igjen med en oppfatning om å ha kommet frem til et produkt som tilfredsstillende oppdragsgivers krav, i tillegg til å ha tilfredsstillende våre forventninger til oppgaven. Spesielt spennende er det at Oppland Arbeiderblad og kanskje andre aviser under A-Pressen kommer til å bruke vår applikasjon. Dette vil være en utmerket referanse å vise til. Hele prosessen har gitt oss flere allsidige og nyttige erfaringer som vil komme til nytte senere i livet.

Definisjoner og ordforklaringer

Administrator

Person/rolle som står for styringen av gratulasjonssystemet. Godkjenner, redigerer og sletter hilsninger via administratorsidene.

Akseptansetest

En test som Payex foretar på gratulasjonssystemet. Hvis systemet godkjennes kan det settes i drift og tjenestene kan bli betalt gjennom Payex sitt system.

A-pressen

Et mediekonsern med virksomhet i over 40 lokalaviser.

A-pressen interaktiv

Avdeling av A-pressen. Fellesorgan for nettutgaven av avisene i A-pressen.

ASP.NET

Microsofts utviklingsplattform som vi har brukt i prosjektet.

Browser

Nettleser for navigasjon på internett, for eksempel Internet Explorer.

Bruker

Person som tar i bruk gratulasjonsløsningen.

C#

Programmeringsspråk som kan brukes i ASP.NET

Cookie

En fil som lagres på internettbrukerens harddisk som brukes til å lagre data om brukeren.

CSS

Cascade style sheet. Et stilsett som gjør det enkelt å implementere likt design og utseendemessige oppsett på flere forskjellige internettsider.

Dpi

Er *dots per inch* eller prikker per tomme. Brukes i forbindelse med oppløsning på et bilde.

Escenic

Nettdesksystemet til Oppland Arbeiderblad.

Filadresse

Adressen til en spesifikk fil på en logisk stasjon. Eksempel: "c:\bilder\mittbilde.jpg"

Flerbrukersystem

Et system som enkelt kan implementeres hos flere aktører.

Hardware

Fysiske komponenter. I vårt tilfelle kan dette være snakk om datautstyr.

HiG

Høgskolen i Gjøvik.

Hilsen

Resultatet av informasjonen bruker legger inn blir en hilsen. Denne består av bilde, tekst og informasjon om bruker/gratulant.

Iframe

Et "vindu" på en nettside der informasjon hentes fra et annet sted på nettet.

Inkrement

En ferdig del av et produkt som kan kjøres, testes og står på egne bein.

MD5-Hash

128 bits krypteringsalgoritme for autentisering over internett.

MySQL – Database

En database er en logisk og strukturert samling av data. SQL står for Structured Query Language og er en standard for å gjøre spørringer mot en database for å finne data. MySQL er gratis programvare som gjør at man kan få sette opp sin egen SQL server.

OA

Forkortelse for Oppland Arbeiderblad. Nettutgaven blir referert til som oa.no.

ODBC

Driver for å gjennomføre kommunikasjon mellom Visual Studio .NET og en MySQL-database.

Open Source

Gratis program utviklet av brukere over hele verden.

Payex

Betalingsystem som kan integreres i internettsider som selger varer og tjenester.

Path

Samme som filadresse.

Piksel

Et punkt i et bilde. Et bilde består av et visst antall piksler horisontalt og et antall piksler vertikalt. Dette gir en oppløsning, for eksempel 400 x 400 piksler.

Pop-up vindu

Nytt nettleservindu som blir åpnet av systemet. I gratulasjonssystemet brukes det blant annet når brukeren ber om å få opp en utskriftsvennlig versjon av en hilsen.

Query

Spørring om data mot SQL-database.

Querystring

En spørrestreng som gjør det mulig å sende informasjon mellom sider i applikasjonen. Eks:
<http://www.dittdomene.no/default.aspx?ID=3>

Saxotech

Oppland Arbeiderblads system for layout/generering av papirutgaven.

SMS

Forkortelse for Short Message Service. Korte meldinger man kan sende vha. mobiltelefoner.

SOAP

Protokoll. Forkortelse for Simple Object Access Protocol. Språk for kommunikasjon mellom bruker og web service.

SQL

Språk for kommunikasjon og bruk av database.

SSL

Secure socket layer. For sikkerhet i dataoverføringer. I HTML blir alle data overført i klartekst. Ved bruk av SSL blir dataoverføringer kryptert, særlig aktuelt ved autentisering og transaksjoner.

Thumbnail

Den lille versjonen av bilde som blir lagret i systemet. Typisk brukt der man skal vise en oversikt over flere hilsninger.

VG

Verdens Gang. I vår rapport brukes VG i forbindelse med vg.no som er nettutgaven av Verdens Gang.

Visual Studio .NET

Programmeringsverktøy laget av Microsoft. Brukt til å utvikle gratulasjonssystemet.

Webhuset.no

Selskap som driver webhotell. Kontor i Bergen.

Webservice

En webservice er en liten programvare som går på en internettsjerver og utfører helt bestemte operasjoner. Disse kan brukes fleksibelt på tvers av nettstedet og andre tjenester.

XML

Et språk for å vise dokumenter som inneholder strukturert tekst/innformasjon.

Litteratur

Skriftlig litteratur:

- Sams: ASP.NET – Kick Start.
Stephen Walter, 2003
- Sams: – Teach your self Visual Studio .NET 2003 in 21 days.
Jason Beres, 2003.
- Beginning .Net Web Services using c#.
Joseph Bustos og Karli Watson, 2002.
- Software Engeneering 6th edition.
Ian Sommerville, 2001.
- På nett med ungdom – A-pressen interaktiv.
Hovedprosjekt ved HiG, 2003.
- Rapportmaler, 2004.
Hovedprosjekt ved HiG, 2004.
- Dynamisk internettløsning for Raufoss industripark.
Hovedprosjekt ved HiG, 2004.
- Payex – Implementeringsguide og retningslinjer for Payex,
Versjon 1.2.
Payex, 2003.

Fra Internett:

- www.asp.net
Generell info om ASP.NET.
- www.payex.com
Betalingssystem.
- www.vg.no/hurra/
Brukt som eksempel.
- www.mysql.com
Sql-koder, nedlasting av server og MyODBC.

- www2.hig.no/at/data/sysutv1/
Faget Systemutvikling sine hjemmesider.
- www.411asp.net
Forum.
- www.dotnetbips.net
Tips og triks.
- www.w3schools.com
Webservice / SOAP.
- www.codeproject.com
Forum, eksempler og info asp.net.
- www.csharpcorner.com
Litt om asp.net.
- www.exif.org
Metadata informasjon i JPEG og TIFF.
- www.msdn.com
Microsoft sine hjelpesider.
- www.stardeveloper.com
Kode eksempler asp.net