



DAIM

Digital **A**rkivering og **I**nnlevering av **M**asteroppgaver

Et informasjonssystem for studenter og administrasjonens arbeid med masteroppgaver.

Student

DAIM skal støtte studenten fra uttak av masteroppgaven, til den er ferdig innlevert.

Kontrakter

Studentene bruker systemet for å fylle ut kontrakter og skjema (masterkontrakt, samarbeidskontrakt og innleveringsskjema). Når nødvendige opplysningene er utfyllt kan studenten skrive ut ferdig utfylte kontrakter og skjema, som bare trenger signering.

Samarbeidsgrupper kan opprettes på nett. To (eller i særskilte tilfeller tre) studenter kan samarbeide om å skrive masteroppgave. Etter at studentene har godkjent at de skal samarbeide i gruppe, kan de skrive ut **samarbeidskontrakt** via DAIM.

Etter innlevering av masteroppgaven via DAIM kan ferdig utfyllt **innleveringsskjema** skrives ut fra DAIM.

Lever kontrakter og skjema i instituttresepsjonen.

Digital innlevering av masteroppgaver

Masteroppgaven innleveres digitalt som *en* PDF fil. Det er viktig at PDF filen har fonter inkludert. Inkluderte fonter forhindrer bytte av font dersom en font brukt i dokumentet ikke finnes på skriveren til Tapir. Digitale vedlegg (kildekode, simuleringer, animasjoner, kjørende program og lignende) kan leveres som *en* ZIP fil.

Etter innlevert masteroppgave skal studenten levere *innleveringsskjema* ferdig signert i instituttresepsjonen.

Omtrent to arbeidsdager etter at masteroppgaven er innlevert, vil den ferdig trykte papirutgaven leveres til instituttresepsjonen. Ekstrabestillinger som studenten kan ha gjort ved innlevering, vil bli levert til Tapir bokhandelen studenten spesifiserte.

Administrasjon

DAIM er et verktøy for administrasjonen av masteroppgaver. Verktøyet bør benyttes både ved institutt og fakultet.

Et overblikk over hva de forskjellige administrasjonsrollene gjør i systemet:

- Admin
- Sysadmin
- Økonomi
- Hovedveileder

Økonomi og hovedveileder rollene er de som får fått minst tid til utvikling i denne omgang. Det er mulighet å gjøre mer ut av disse rollene, samt å legge inn en sensor rolle.

Admin

Admin rollen håndterer administrasjon av kontrakter og skjemaer for masteroppgaver, innlegging av studenter som mangler tilgang og vedlikehold av hovedveilederliste og sensorliste. Denne rollen er best for å hente ut oversikter fra systemet.

Masteroppgaver

Studentene fyller selv ut det meste av informasjonen som trengs for administrasjon av oppgavene, slik at administrasjonen må i større grad bare kontrollere at informasjonen er rett utfyllt.

Når det legges inn en start dato for masteroppgaven, og det er valgt et studieprogram, så skal DAIM regne ut hva som blir innleveringsfristen. Administrasjonen har mulighet til å endre på denne fristen.

Administrasjonen kan endre status for studenter, samt legge inn *merknader* for hver masteroppgave og hver student. Det er også mulig å legge inn båndlegging for hver oppgave.

Både hovedveiledere og administrasjon kan legge inn sensor for oppgaver. Hovedveiledere kan bare velge blant ordinære sensorer, mens administrasjonen kan legge inn både ordinære og ekstraordinære sensorer for en oppgave.

Informasjon om: hvilken sensor, sensurfrist, når en oppgave er sendt til sensur, når den er mottatt, og hvilken karakter masteroppgaven fikk blir lagret for hver oppgave.

Både institutt og fakultet må inn å behandle masteroppgavene.

Hovedveiledere

Institutt administrasjonen er ansvarlig for å holde informasjonen om hovedveiledere ved instituttet oppdatert. Listen inneholder fast ansatte hovedveiledere ved instituttet, og studentene bruker denne listen for å velge hovedveileder til masteroppgaven.

Personer som har veiledet en eller flere oppgaver kan ikke slettes fra listen, men de kan settes til **inaktiv** som betyr at de ikke lenger veileder oppgaver (da kan ikke studenter velge dem som hovedveileder lenger).

Sensorer

Institutt administrasjonen er ansvarlig for å holde informasjonen om instituttets sensorer oppdatert.

Informasjon om sensor skal legges inn i når man sender forespørsel om oppnevning av sensor til fakultetet. Dermed blir en kopi av opplysningene lagt inn, og man trenger ikke å huske på å oppdatere dette senere.

Sysadmin

Sysadmin har tilgang til de samme funksjonene som admin, men har utvidede rettigheter. De kan importere data fra FS for studenter og sensorer, samt endre på informasjon for institutt, og administrere admin brukere.

NB! Det skal være en lokal sysadmin ved hvert institutt. Lokal sysadmin for et institutt er ansvarlig for import av studenter, samt administrasjon av administratorroller ved det instituttet.

Utvalgte sysadmin har utvidede rettigheter til å simulere andre brukere, samt endre på opplysninger for opptrykk av masteroppgaver.

Økonomi

Økonomi rollen skal håndtere alt som har med betaling for masteroppgaver å gjøre.

Mangler Honorering

Ubetalte oppgaver som er levert og har fått satt sensor og karakter, vil vises i en liste over masteroppgaver som mangler sensurhonorar.

Betalte Honorar

Oppgaver som har fått betalt honorar vises i en egen side. Innleverte oppgaver skal finnes i ei av de to listene.

Hovedveileder

Hovedveileder skal kunne få ut en oversikt over hvilke oppgaver de har veiledet som hovedveileder.

Mine masterstudenter

Oversikten visere hvilke masterstudenter som skriver eller har skrevet for veilederen. Hvilken sensor som er valgt for hver oppgave. Om en oppgave har fått sensur, og når den ble innlevert. Hovedveileder skal kunne velge sensor for oppgaver som ikke har fått sensur.

Mine oppgavetekster

Dette skal være en side der hovedveiledere legger ut forslag til masteroppgaver som de ønsker at studenter skal ta. Studentene skal kunne lete frem i beskrivelsene via søking i studentgrensesnittet. *(Ikke implementert)*

Søkeside for DAIM

Lagring og gjenfinning av digitale masteroppgaver er et av hovedmålene for DAIM systemet. DAIM lagrer beskrivelsesdata/metadata, samt de digitalt innleverte masteroppgavene slik at man senere kan gjøre disse tilgjengelige for søking.

Tittelsiden skal inkluderes automatisk *først i hver masteroppgave* som leveres, slik at tittel, forfatter, veileder og NTNU logo er synlig på første side av hver masteroppgave.

Det skal være mulig å søke etter "alle" masteroppgaver som er levert i DAIM via søkegrensesnittet. Søking etter dokumenter skal ikke kreve noen form for innlogging.

Masteroppgavene som blir vist skal:

- Ha fått sensur
- Student skal ha godtatt publisering
- Ikke være båndlagt

Søkesiden er viktig for å vise forskningen som gjøres ved NTNU, og hvilke fagretninger forskjellige veiledere er villige til å veilede for.

Søkesiden vil kunne bli NTNU sitt vindu mot omverdenen. En kilde til inspirasjon og kunnskap for informasjonssøkere.

Tekniske aspekter

Dette kapitlet tar for seg de tekniske aspektene ved DAIM systemet.

Felles database

All informasjonen som blir lagret i DAIM skal legges inn i en felles database. Denne databasen skal senere kunne gjøres bruk av for andre programmer. For eksempel kan det utvikles programmer som regner ut gjennomsnitt karakter som hver sensor gir, og da kan man få ut informasjon om hvilke sensorer som er for ”snille” og hvilke som er for ”strengt”.

I størst mulig grad bør felles datalager benyttes, slik at alle endringer bare må gjøres *en* gang og *et* sted. Dersom alle benytter *ett* felles datalager kan man være sikker på at informasjonen som ligger lagret der er den mest oppdaterte.

Sentrale aktører i DAIM

En aktør i et system kan være enten menneskelig eller et eksternt system som fungerer som en aktør i forhold til DAIM systemet.

Administrasjon

- Systemadministrator med utvidet tilgang
- Administrasjon
- Økonomi
- Hovedveileder
- Sensor? (mulig utvidelse senere)

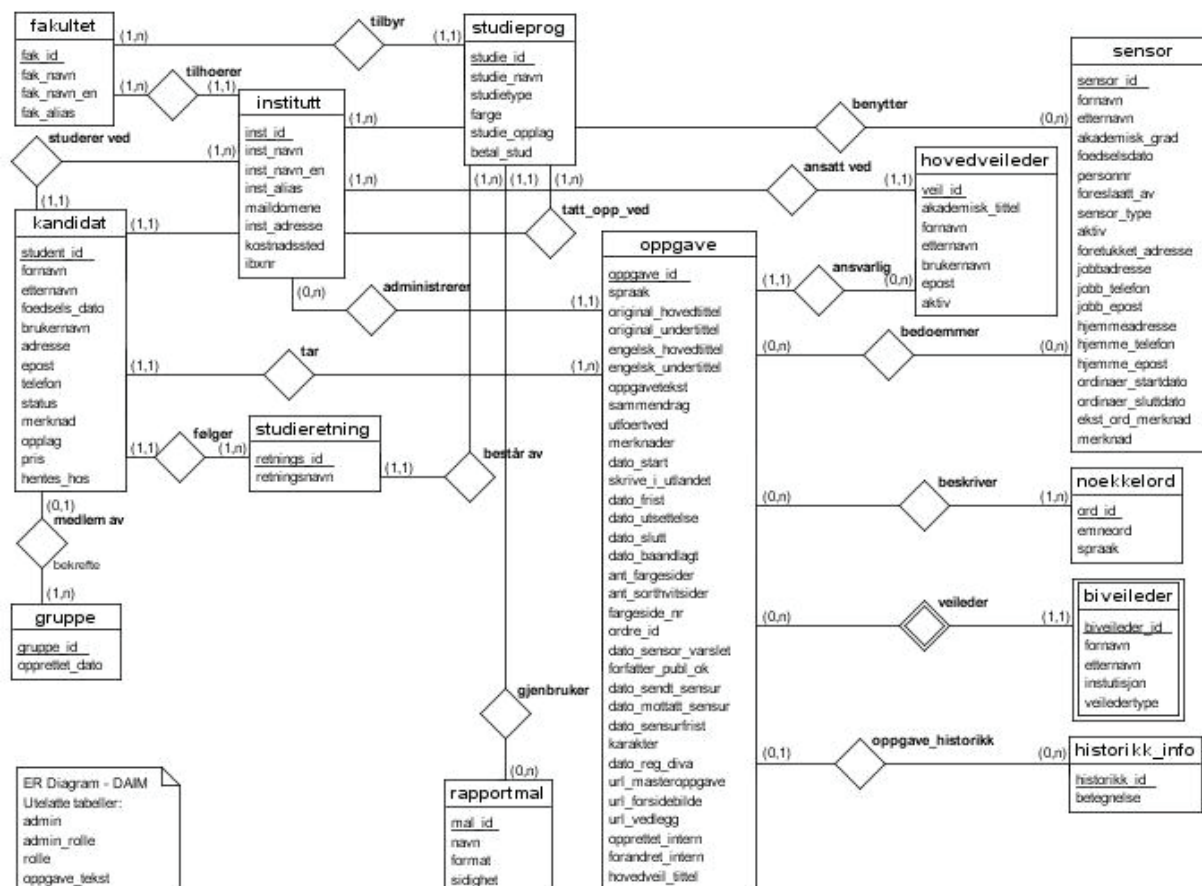
Student som benytter systemet som er verktøy for å skrive masterkontrakt, levere masteroppgaven digitalt og avslutte masterstudiet.

Informasjonssøker som bruker søkesidene i DAIM for å finne informasjon og kunnskap.

Tapir Uttrykk sitt system vil ikke kontakte DAIM systemet direkte. Det vil kun sende respons tilbake når det får XML meldinger fra DAIM systemet. DAIM ”pusher” informasjon til Tapir Uttrykk ved behov, og da vil man få respons meldinger tilbake.

XML og XML schema benyttes for å definere protokollene for kommunikasjon mellom Tapir og DAIM systemene.

ER diagram for databasen



Forklaring av tabeller og attributter

Attributter:

Kursiv = fremmednøkkel

Understreking = hovednøkkel

admin

Table comments: Brukere med adminroller.; InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>auth_id</u>	int(11)	No	NULL	Primærnøkkel
fornavn	varchar(25)	No		
etternavn	varchar(25)	No		
brukernavn	varchar(15)	No		
epost	varchar(25)	No	NULL	
<i>default_inst</i>	tinyint(4)	No		Hvilket institutt man jobber hovedsaklig med

admin_rolle

Table comments: Hvilke roller hver admin har.; InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>auth_id</u>	int(11)	No		
<u>rolle_id</u>	smallint(6)	No		
readonly	tinyint(4)	Yes	0	Bare lesetilgang (benyttes ikke enda)
kommentar	varchar(50)	Yes	NULL	
inst_id	int(11)	No	0	For hvilke(t) institutt
fak_id	tinyint(4)	No	0	For hvilke(t) fakultet

akademisk_tittel

Table comments: Akademist tittel for hovedveiledere InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>tittel_id</u>	int(11)	No	0	Primærnøkkel
tittelnavn	varchar(40)	No		Navn på tittel (Professor, Førsteamanuensis)

auth

Table comments: Lokal autentisering.; InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>user</u>	varchar(15)	No		Brukernavn
pw	varchar(35)	No		Password
created	datetime	Yes	NULL	Opprettet
changed	datetime	Yes	NULL	Forandret
expire	datetime	Yes	NULL	Utgår
kommentar	varchar(200)	Yes	NULL	

bedoemmer

Table comments: Hvilke oppgaver sensor bedømmer. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>sensor_id</u>	int(11)	No	0	
<u>oppgave_id</u>	int(11)	No	0	

benytter

Table comments: Hvilke sensorer et institutt benytter. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>inst_id</u>	int(11)	No	0	
<u>sensor_id</u>	int(11)	No	0	

beskriver

Table comments: Hvilke ord som beskriver en oppgave. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>ord_id</u>	int(11)	No	NULL	
<u>oppgave_id</u>	int(11)	No	NULL	

biveileder

Table comments: Biveileder for masteroppgave InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>biveileder_id</u>	int(11)	No	NULL	Primærnøkkel
<u>oppgave_id</u>	int(11)	No	0	Primærnøkkel
fornavn	varchar(25)	No		
etternavn	varchar(25)	No		
institusjon	varchar(80)	No	IDI	Instituttalias eller bedriftsnavn
veiledertype	enum('intern', 'ekstern')	No	intern	Ekstern eller intern biveileder

fakultet

Table comments: InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>fak_id</u>	int(11)	No	0	Primærnøkkel
fak_navn	varchar(80)	No		Norsk (bokmål) Fakultetsnavn
fak_navn_en	varchar(80)	No		Engelsk fakultetsnavn
fak_alias	varchar(15)	No		Fakultetets alias (f.eks. IME)

gruppe

Table comments: InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>gruppe_id</u>	int(11)	No	0	Primærnøkkel
opprettet_dato	date	No	0000-00-00	Når gruppe ble opprettet

historikk_info

Table comments: Inneholder gamle betegnelser for instituttet. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>historikk_id</u>	int(11)	No	0	Primærnøkkel
betegnelse	varchar(255)	No	NULL	Gamle instituttalias

hovedveileder

Table comments: InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>veil_id</u>	int(11)	No	0	Primærnøkkel
<u>tittel_id</u>	int(11)	No	0	Akademisk tittel
fornavn	varchar(25)	No		
etternavn	varchar(25)	No		
brukernavn	varchar(15)	Yes	NULL	
epost	varchar(80)	Yes	NULL	
aktiv	tinyint(1)	No	0	Kan veilede masteroppgaver? (ja/nei)
<u>inst_id</u>	int(11)	No	0	Hvilken institutt hovedveileder tilhører

institutt

Table comments: InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>inst_id</u>	int(11)	No	0	Primærnøkkel
inst_navn	varchar(80)	No		Norsk institutt navn (bokmål)
inst_navn_en	varchar(80)	No		Engelsk institutt navn
inst_alias	varchar(15)	No		Institutt alias
maildomene	varchar(25)	Yes	NULL	f.eks. idi.ntnu.no
inst_adresse	varchar(80)	Yes	NULL	Postadresse for instituttet
kostnadssted	int(8)	Yes	NULL	Kostnadssted for instituttet f.eks. 631005 for IDI
fak_id	int(11)	No	0	Hvilken fakultet instituttet tilhører
ibxnr	varchar(10)	Yes	NULL	IBXnr for bestilling av masteroppgaveopptrykk

kandidat

Table comments: Studentinformasjonen legges inn her. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
student_id	int(11)	No	0	
foedsels_dato	date	No	NULL	
fornavn	varchar(25)	No		
etternavn	varchar(25)	No		
brukernavn	varchar(15)	No	NULL	
status	enum('registrert', 'kontrakt_ok', 'permisjon', 'avbrutt', 'ferdig', 'karakter')	No	NULL	Status for arbeidet (registrert, kontrakt_ok, permisjon, avbrutt, ferdig)
merknad	text	Yes	NULL	
adresse	tinytext	Yes		Adresse for tilsendning av vitnemål
telefon	varchar(35)	Yes	NULL	
epost	varchar(80)	Yes	NULL	
opplag	int(3)	Yes	0	Antall trykte kopier student har bestilt
pris	int(5)	Yes	0	Pris for kopiene student har bestilt
hentes_hos	varchar(25)	Yes	NULL	Hvor kopiene skal hentes
inst_id	int(11)	Yes	0	Studentens institutt
studie_id	varchar(12)	Yes	0	Studentens studieprogram
retnings_id	int(11)	Yes	0	Studentens studieretning
oppgave_id	int(11)	Yes	0	Oppgaven til studenten.
innlagt	date	Yes	NULL	
endret	datetime	Yes	NULL	
v_tittel	tinyint(4)	Yes	NULL	
endret_av	varchar(15)	Yes	NULL	

medlem_av

Table comments: Hvilke studenter som utgjør en gruppe. NB! Hvis en av studentene *avkrefter* samarbeid, vil hele gruppen termineres.

InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>gruppe_id</u>	int(11)	No	0	
<u>student_id</u>	int(11)	No	0	
bekreft	tinyint(1)	Yes	NULL	

noekkelord

Table comments: Termer som kan brukes for å beskrive en oppgave. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>ord_id</u>	int(11)	No	0	
emneord	varchar(30)	No		
spraak	char(2)	No	NB	Ett av de fem NTNU språkene (NB,NN,EN,DE,FR)

oppgave

Table comments: Hovedtabellen for hver masteroppgave. NB! Hvis ingen studenter er knyttet til en oppgave, så skal oppgaven slettes fra databasen. Viktig å teste på når man sletter fremmednøkkel for *oppgave_id* i kandidat tabellen. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>oppgave_id</u>	int(11)	No	0	Primærnøkkel
spraak	char(2)	No	NB	Hvilket språk oppgaven er skrevet på
original_hovedtittel	varchar(255)	Yes		Oppgavetittelen på [spraak]
original_undertittel	varchar(255)	Yes	NULL	Oppgaveundertittel på [spraak]
engelsk_hovedtittel	varchar(255)	Yes		Engelsk hovedtittel for oppgaven
engelsk_undertittel	varchar(255)	Yes	NULL	Engelsk undertittel for oppgaven
oppgavetekst	text	Yes	NULL	Problembeskrivelse
sammendrag	text	Yes	NULL	Oppsummering av de viktigste sidene
utfoertved	varchar(80)	Yes		Skrevet ved inst., bedrift eller i utlandet
merknader	text	Yes	NULL	Fra student eller administrasjon
dato_start	date	Yes	NULL	Dato for oppstart av oppgaven
skrive_i_utlandet	tinyint(1)	Yes	0	Skrives i utlandet? (Ja/Nei)
dato_frist	date	Yes	NULL	Innleveringsfristen
dato_utsettelse	date	Yes	NULL	Utsettelse til dato
dato_slutt	date	Yes	NULL	Datoen da oppgaven ble levert
dato_baandlagt	date	Yes	NULL	Dato for når båndleggingen oppheves
baandlagt_valg	tinyint(4)	Yes	NULL	Valg av tidsperiode for båndlegging
ant_fargesider	smallint(4)	Yes	0	Antall fargesider som skal trykkes
ant_sorthvitsider	smallint(6)	Yes	NULL	Antall sort/hvitsider som skal trykkes
fargesidenr	varchar(50)	Yes	NULL	Sidetall for fargesider som skal trykkes
ordre_id	varchar(50)	Yes	NULL	Tapir's ordreid for bestilling av opptrykk
trykkepris	float	Yes	0	Pris for opptrykk fra Tapir
forfatter_publ_ok	tinyint(1)	Yes	0	Skal oppgaven publiseres? (Ja/nei)
dato_sensor_varslet	datetime	Yes	NULL	Dato sensor ble varslet (ikke brukt enda)
dato_sendt_sensur	date	Yes	NULL	Dato sensur ble sendt til sensor
dato_mottatt_sensur	date	Yes	NULL	Dato sensur ble mottatt fra sensor
dato_sensurfrist	date	Yes	NULL	Dato for sensurfrist
karakter	varchar(40)	Yes	NULL	Karakteren oppgaven fikk
dato_reg_diva	date	Yes	NULL	Dato for registrering i Dig. Bibliotek
url_masteroppgave	varchar(80)	Yes	NULL	URL til masteroppgaven
url_forsidebilde	varchar(80)	Yes	NULL	URL til forsidebilde

url_vedlegg	varchar(80)	Yes	NULL	URL til vedlagt ZIP fil
opprettet	datetime	Yes	NULL	Dato for når oppgaven ble opprettet
endret	datetime	Yes	NULL	Dato for siste endring av oppgaven
inst_id	int(11)	Yes	0	Hvilket institutt oppgaven tilhører
veil_id	int(11)	Yes	NULL	Hvem som er hovedveileder
hovedveil_tittel	varchar(25)	Yes	NULL	Tittel hovedveileder hadde ved veiledning
pdf_sjekksum	varchar(35)	Yes	NULL	PDF sjekksum (MD5 av opplastet fil)
sensur_honorar	tinyint(4)	Yes	0	Er sensurhonorar betalt? (ja/nei)
endret_av	varchar(15)	Yes		Siste person som gjorde endringer

oppgave_historikk

Table comments: Oppgaver som var knyttet gamle institutt alias. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>historikk_id</u>	int(11)	No	0	
<u>oppgave_id</u>	int(11)	No	0	

oppgave_tekst

Table comments: Hvilke oppgavetekster hovedveileder tilbyr. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>oppgavetekst_id</u>	int(11)	No	0	
<u>veil_id</u>	int(11)	No	0	
hovedtittel	varchar(180)	No		
undertittel	varchar(180)	Yes	NULL	
oppgavetekst	text	No		
biveileder	varchar(180)	Yes	NULL	
studietype	enum('alle', 'sivilingeniør', 'allmennvitenskapelig')	No	alle	
aktiv	tinyint(4)	Yes	0	Mulig å velge nå? (ja/nei)
endret_dato	date	Yes	0000-00-00	

rapportmal

Table comments: Hvilken mal som skal brukes for trykking. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>mal_id</u>	int(11)	No	0	
mal_navn	varchar(30)	No		
format	varchar(20)	No		A4 normalt sett
sidighet	tinyint(4)	No	0	Tosidig eller ensidig layout

rolle

Table comments: Hvilke forskjellige roller som finnes for admin. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
<u>rolle_id</u>	tinyint(4)	No		
rolle_navn	varchar(25)	No		Admin, Sysadmin osv.

sensor

Table comments: InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
sensor_id	int(11)	No	0	
fornavn	varchar(25)	No	NULL	
etternavn	varchar(25)	No	NULL	
akademisk_grad	varchar(30)	Yes	NULL	
foedselsdato	date	Yes	NULL	
personnr	int(11)	Yes	NULL	
foreslaatt_av	varchar(50)	Yes	NULL	Hvilken veileder ønsket sensor oppnevnt?
sensor_type	tinyint(4)	No	0	Ordinær eller Ekstraordinær?
aktiv	tinyint(4)	No	0	Kan sensurere for øyeblikket?
foretrukket_adresse	tinyint(4)	No	1	Sende info til jobb eller hjemme adresse?
ordinaer_startdato	date	Yes	NULL	Start av ordinær periode
ordinaer_sluttdato	date	Yes	NULL	Slutt av ordinær periode
ekst_ord_merknad	text	Yes	NULL	Ekstraordinær merknade
merknad	text	Yes	NULL	
jobbadresse	varchar(100)	Yes	NULL	
jobb_telefon	varchar(35)	Yes	NULL	
jobb_epost	varchar(80)	Yes	NULL	
hjemmeadresse	varchar(100)	Yes	NULL	
hjemme_telefon	varchar(35)	Yes	NULL	
hjemme_epost	varchar(80)	Yes	NULL	

studieprog

Table comments: Hvilke studieprogram som tilbys ved fakultet. InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
studie_id	varchar(12)	No		
studie_navn	varchar(80)	No		Norsk (bokmål) navn på studieprogram
studie_navn_en	varchar(160)	No		Engelsk navn på studieprogram
studietype	varchar(25)	No		Allmennvitenskapelig eller sivilingeniør
farge	varchar(15)	No		Hvilken farge skal brukes ved trykking?
mal_id	int(11)	No	0	Hvilken rapportmal brukes ved trykking
fak_id	int(11)	No	0	Hvilket fakultet tilhører programmet?
studie_opplag	smallint(10)	No	3	Hvor mange opplag skal trykkes?
betal_stud	tinyint(2)	No	1	Får hver student ett opplag ved opptrykking?
studie_aktiv	tinyint(2)	No	1	Tilbys stuideprogrammet for øyeblikket?

studieretning

Table comments: Hvilke studieretninger som finnes. Hver retning er knyttet til et studieprogram.
InnoDB free: 41347072 kB

Field	Type	Null	Default	Comments
retnings_id	int(11)	No	0	
retningsnavn	varchar(60)	No		Navnet på studieretningen
studie_id	varchar(12)	No		Studieprogram retningen er tilknyttet
retning_aktiv	tinyint(2)	No	1	Tilbyes retningen for øyeblikket? (ja/nei)

Forklaring til ER diagram for DAIM

Tabellene *admin*, *admin_rolle*, *rolle*, *auth* og *oppgave_tekst* er med vilje utelatt fra oversikten på ER diagrammet. De tre første brukes i forhold til autentisering av administratorer.

Auth brukes for å legge inn brukere som bare skal ha lokal tilgang til DAIM systemet uten å legges inn på innsida (testbrukere og demobrukere).

Oppgave_tekst tabellen kommer til å benyttes for å legge inn oppgavetekster som hovedveiledere vil tilby til studenter. Denne vil bli implementert ved en senere oppdatering av DAIM systemet.

Hovedtabellen i databasen er oppgavetabellen. En oppgave kan ha en eller flere studenter tilknyttet seg. Hver student kan jobbe med en masteroppgave.

Forbedringspotensiale:

1. En studieretning kan finnes ved flere studieprogram.
2. Et studieprogram kan være knyttet til flere fakultet.
3. Institutt utenom fakultetet som primært tilbyr studieprogrammet, skal kunne tilby studieprogrammet.

Historikk

Det er mulig å legge inn mer historikk i databasen. Historikk kan fort føre til at databasen blir kompleks.

Mulig historikk kan være:

- Hvilke **sensorperioder** en sensor har jobbet.
 - Må ta høyde for at sensor kan begynne midt i en periode.
- **Fra – Til** for Hovedveileder, sensor, studieprogram, studieretninger, administratorer osv.
 - Legge inn "opprettet" attributt som *fra* i alle tabeller.
 - Når status settes til *inaktiv* lagres en *til* dato i tabellen.
- **Karakterer** ved klagesaker, se mulig løsning under.

Karakterer og klagesaker

Det registreres karakter og sensor for oppgave.
Hvis det blir klagesak vil oppgaven få en ny karakter.

Når karakter legges inn i *oppgave* tabellen, legges den inn både som *original karakter* og som *endelig karakter*. Hvis oppgaven får en klage og gis ny karakter, legges ny karakter inn i *endelig karakter* feltet.

Hva oppnår vi med dette?

- Original karakter vil vise karakteren som sensor har gitt for oppgaven
 - o Mulig å kalkulere karakter for hver sensor/hovedveileder korrekt
- Endelig karakter vil være karakteren student ser
- Hvis original og endelig karakter er forskjellig betyr det at det har vært en klagesak
 - o Vil ikke vises hvis klage gav samme karakter som originalt.
 - o Vil kunne vise hvor mange som endrer karakter ved klage

I følge Bård Kjos gjelder dette såpass få personer hvert år at det er neglisjerbart. Derfor har vi kun ett felt for karakter i databasen.

Samarbeid mellom studenter med forskjellige studieprogram

Studenter med samme studietype kan samarbeide om en masteroppgave. Sivilingeniører og allmennvitenskapelige studenter kan ikke samarbeide om en masteroppgave p.g.a. forskjellig tidsperspektiv for masteroppgaven.

Studenter fra samme studietype men med forskjellig studieprogram kan samarbeide om å skrive masteroppgave. For eksempel sivilingeniører fra datateknikk linja ved IDI kan samarbeide med sivilingeniører fra andre institutt og fakultet.

Studieprogrammet for en oppgave må lagres som studieopplysning for hver student som skriver oppgaven. På denne måten kan en oppgave bli skrevet ved flere forskjellige program samtidig.

Dette må taes hensyn til når man skriver ut kontrakter, **samt ved generering av forside og tittelside når det skal trykkes ved trykkeriet.**

Begrensninger i DAIM

Felles karakter for samarbeidsgruppe

DAIM støtter kun felles karakter når en masteroppgave leveres som gruppesamarbeid. Individuelle karakterer for gruppesamarbeid støttes **ikke** av DAIM. De siste årene er det ingen ved IDI som har ønsket individuelle karakterer for gruppesamarbeid.

DAIM er **ikke** et system for lagring av den endelige karakteren (FS er systemet for karakterer). Grunnen til at karakter er del av DAIM systemet, er i forhold til kvalitativ utvelgelse av oppgaver som kan publiseres i eksterne digitale bibliotek. En regel som ble foreslått av Kjell Bratbergsengen (Instituttleder ved IDI) er at *alle oppgaver som har fått karakteren C eller bedre, er oppgaver som man ønsker publisert.*

Selv om kun en karakter lagres for hver masteroppgave, er det felt for avkrysning på samarbeidsskjema der man kan velge individuell karakter. Dette gjør at man støtter individuelle karakterer i forhold til administrasjon på kontrakter, men man kan kun lagre *en* karakter for oppgaven i DAIM. **Problem i forhold til visning av karakter for oppgave, samt for bruk av karater for publisering?**

Studentsamarbeid mellom to forskjellige institutt/fakultet

Dersom to studenter fra forskjellige institutt ønsker å samarbeide om en masteroppgave, er de nødt til å levere ved ett av instituttene.

Utfordringer i forhold til samarbeid av studenter fra forskjellige institutt:

- Hvilket av instituttene skal ha hovedveileder for oppgaven?
 - o Det instituttet hvor oppgaven i praksis gjennomføres
- Hvilket av instituttene skal oppnevne sensor og betale for sensur?
 - o Det instituttet som har ansvar for administrasjon av oppgaven.
- Hvilket instituttnavn skal stå på forsiden av den trykte masteroppgaven? Det er plass til bare ett fakultetsnavn, og ett instituttnavn.
 - o **Ingen enkel løsning.**
- Hvem skal ha honorar for veiledning?
 - o Det instituttet der hovedveileder jobber ved.

Bård Kjos har gitt svarene over.

Siden hovedveileders institutt får honoraret for veiledning, bør også det instituttet dekke kostnader for sensur og gjøre oppnevningen og administrasjonen av masteroppgaven. Det er rettferdig at **kostnader og inntekter går til samme institutt.**

Antall eksemplarer som trykkes for hvert studieprogram blir gitt av studieprogrammet og ikke instituttet. Dette er for at studenter fra forskjellige institutt som går samme studieprogram skal behandles likt.

Forskerskolen

Hvert år tas det opp fire studenter fra sivilingeniørstudiet til *forskerskolen*. Disse fire personene får to semester på å gjennomføre en forskningsplan (godkjennes som masteroppgave). Dermed får disse studentene 40 i stedet for 20 uker på å gjøre ferdig masteroppgaven.

I merknadsfeltet for student, kan studenten legge inn meldinger som påvirker innleveringsfristen. Derfor skal bare studenten legge inn merknad om forskerskole i merknadsfeltet, slik at administrator kan ta høyde for dette når datoene for uttak og innlevering skal godkjennes.

Utsatt uttak

Ved utsatt uttak skal studenten gi beskjed om utsatt uttak via meldingsboksen i studentgrensesnittet. Hovedveileder skal godkjenne utsatt uttak, og masterkontrakten skal leveres på normal måte innen normal frist.

DAIM har ikke noe eget felt for registrering av utsatt uttak, men her brukes merknadsfeltet på samme måten som ved forskerskolen eksemplet over.

Historiske administrative data

DAIM systemet er **ikke** et system som hovedsakelig er laget for å ta vare på historiske administrative data. Hovedfokuset til DAIM er administrasjon av masteroppgaver, digital innlevering av masteroppgaver, samt søking i lagrede masteroppgaver.

Hvilke historiske administrative data er det som **ikke** lagres?

- Informasjon om tidligere perioder en sensor har arbeidet. DAIM er interessert i om en sensor har mulighet til å sensurere oppgaver for øyeblikket.

Hvorfor lagres ikke denne historiske informasjonen? DAIM er et system som kan gripe inn i mange områder innenfor administrasjonen ved instituttene og fakultetet. For å holde designet av databasen og systemet oversiktlig og forståelig har vi valgt å sette en del grenser for hva DAIM skal kunne gjøre.

DAIM er et system som skal ha fokus på virkeligheten slik den er for øyeblikket. Historikk i stor skala kompliserer både databasedesignet, samt brukergrensesnittet.

Språk på omslaget

Norsk (bokmål og nynorsk) vil få måneds-, institutt- og fakultetsnavn på bokmål. Engelske, franske og tyske oppgaver vil få navnene på engelsk.

Oversikt - filer

Organisering av filer i DAIM ut fra rotkatalogen (**htdocs/**):

Katalog	Innhold
/	Introduksjonen til DAIM - index.php Faq - faq[].php Demo - *.html og *.swf Vise filer for oppgave - show.php
lib/	Klassefiler <ul style="list-style-type: none">• Datoer – dates.php• Autentisering – auth.php• Oppgave – oppgave.php• Kandidat – kandidat.php• Templates – templates.php• Oppdateringer – update.php• Database – db.php• Kommunikasjon – komm.php• XML – xml.php
inc/	Rammeverket for sidene - daim.inc Globale variabler - daim.inc Admin meny - meny.inc
css/	Stilark – student, administrasjon, print
soek/	Søking – Søkegrensesnittet og resultatsider
admin/	Administrasjon – admin, sysadmin, hovedveileder
masterkontrakt/	Min masteroppgave – index.php Hovedkatalogen for student filer.
masterkontrakt/kontrakt/	Skjema og kontrakter genereres ut fra disse filene
bilder/	Bildefiler - illustrasjoner
images/	Bildefiler – stilark og layout
javascript/	Javascriptkode
xml/	XML schemafiler

Tabellen viser en generell beskrivelse over hva katalogene inneholder. Filene for hver masteroppgave blir ikke lagret direkte i html trestrukturen. Dette er for å forhindre at filer kommer på avveie.

Oppgavefilene blir lagret i en katalog kalt *oppgaver/* som ligger på samme nivå som *htdocs/*. Denne katalog er ikke tilgjengelig via web, og man får kun tilgang til filer som ligger i *oppgaver/* via *show.php* som ligger i *htdocs/* katalogen.

Oversikt – klasser

Klasser som benyttes i DAIM finnes i *lib/* katalogen. For å forenkle vedlikehold og gjenbruk av kode er det benyttet klasser med metoder. PHP5 støtter objektorientert programmering.

Klassene blir beskrevet overfladisk med hva klassen sin hovedrolle er og hvilke metoder den innehar.

Auth

Auth klassen brukes for å autentisere (validere om det skal gis tilgang til DAIM) brukere i DAIM systemet.

DAIM bruker innlogging via Innsida, slik at brukere kan gjøre bruk av sitt NTNU brukernavn og passord. Det er også mulig å opprette lokale brukere som kan få tilgang uten å måtte legges inn i NTNUs BDB (Brukerdatabase).

Class: auth.php

Include: db.php
SSOclient.php

Methods:

```
__construct($redirect=null,$target=null)
setAuth()
sjekkHovedveileder()
setRoller($refresh=false)
admin_epost()
minRolle()
is_readonly()
get_inst_id()
rolleListe()
getDefaultRole()
isAdmin()
isSysadmin()
getUser()
```

Db

Databaseklassen håndterer det som har med databasen å gjøre. Brukernavn til databasen ligger i klartekst i denne klassen.

Class: db.php

Include:

Methods:

```
__construct($ro=null,$dbname=null)
addRow($table,$row)
getError()
query($sql)
getRows($query)
showError($sql,$msg='DATABASEPROBLEM')
tableInfo($table=null)
runsql($sql,$rollback=null,$silent=false)
affected()
existRecord($query)
getMax($table,$field)
close()
__destruct()
```

Dates

Dates klassen er laget for å håndtere datoer og konvertering mellom forskjellige datoformater. MySQL bruker kun yyyy-mm-dd formatet når datoer skal lagres og hentes ut, men DAIM bruker datoer i mange forskjellige format. Derfor er det viktig å kunne lett konvertere mellom forskjellige datoformater.

Class: dates.php

Include:

Methods:

```
__construct($date=null,$descr=null,$errURL=null)
convert($date=null,$format='yyyy-mm-dd',$showerrmsg=null)
setErrURL($url)
showError($msg,$url=null)
```

Kandidat

Kandidat klassen representerer en student, og har metoder og verdier som er relevant for en student.

Class: kandidat.php

Include: db.php

Methods:

```
__construct($username=null,$id=null,$debugLevel=null)
setDebugLevel($level=0)
getDebugLevel($level=0)
setInfo()
getInfo()
getStatus()
erMasterstudent($status=null)
sjekkGruppe()
getGruppeMedlemmer()
studinfo($where=null)
erGruppemedlem()
getStudentID()
getGruppe()
erReadOnly()
bekreftGruppe()
slett_gruppe($gid)
slett_fra_gruppe($gid,$student_id=null)
retningsnavn($id=null)
institutt_option_list($fak_id=3,$selected=null)
fakultet_option_list($selected)
js_fakinfo()

showInfo()/* For debugging */
```

Assosiativt array:

```
stdClass Object (
    [student_id] => 2304
    [foedsels_dato] => 1969-03-12
    [fornavn] => Kai Torgeir
    [etternavn] => Dragland
    [brukernavn] => kaitorge
    [status] => ny
    [merknad] => KUN FOR TESTING.
    [adresse] => Moholt Alle 15 - 22 7050 Trondheim
    [telefon] => 94667302
    [epost] => kaitorge@idi.ntnu.no
    [retnings_id] => 2
    [oppgave_id] => 1103
    [innlagt] => 2005-10-15
```

```
[gruppe_id] => 323
[studie_navn] => Master i informatikk
[inst_navn] => Institutt for datateknikk og informasjonsvitenskap
[inst_alias] => IDI )
```

Det assosiative arrayed vist her viser bare noen få verdier som eksempler, og er ikke komplett. Opprett en klasse og skriv den ut for å få se alle verdier som lagres. For eksempel *print_v(\$K,o)* dersom \$K er et instansiert kandidat objekt.

Verdiene for student klassen hentes fra flere tabeller. SQL kode:

```
SELECT kandidat.*, medlem_av.gruppe_id, medlem_av.bekreftede, studieprog.*,
       institutt.*, fakultet.*
FROM kandidat left join studieprog on studieprog.studie_id=kandidat.studie_id
              left join medlem_av on kandidat.student_id=medlem_av.student_id
              left join institutt on institutt.inst_id=kandidat.inst_id
              left join fakultet on institutt.fak_id=fakultet.fak_id
WHERE [betingelse(r)]
```

Komm

Komm klassen brukes for å håndtere XML kommunikasjon mellom DAIM og Tapir Uttrykks trykkerisystem. Klassen benytter seg av CURL for å sende og motta XML.

Class: komm.php
Include: auth.php
 daim.inc
 db.php
 kandidat.php
 oppgave.php
 xml.php

Methods:

```
__construct($url=null)
init($url)
send($type='order',$tileggsdata=null)
set_respons($xmldata=null,$type=null)
get_respons()
lagXML($type=null,$data=null)
```

Oppgave

Oppgaveklassen er hovedklassen i DAIM. En oppgave vil alltid ha en eller flere studenter tilknyttet seg, og en oppgave skal tilhøre ett institutt. Oppgaven tilhører instituttet til hovedveileder for oppgaven, siden både inntekter og utgifter bør følge hverandre.

Class: oppgave.php

Include: db.php

Methods:

```
__construct($kandidat=null,$debugLevel=null)
setNextID()
setDebugLevel($level=0)
getDebugLevel($level=0)
setInfo($id=null)
getInfo()
setOppgaveID()
getStatus()
setStatus()
getHovedveileder()
getBiveiledere()
getSensor()
getMalInfo($mal_id=1)
katalog()

showInfo() /* for debugging */
```

Assosiativt array:

```
stdClass Object (
    [oppgave_id] => 1103
    [spraak] => NB
    [original_hovedtittel] =>
    [original_undertittel] =>
    [engelsk_hovedtittel] =>
    [engelsk_undertittel] =>
    [oppgavetekst] =>
    [sammendrag] =>
    [merknader] =>
    [dato_start] => 2005-11-21
    [dato_frist] =>
    [dato_utsettelse] =>
    [opprettet_intern] => 2005-10-24
    [veil_id] => 19
    [hovedveil_tittel] => Professor )
```

Eksemplet over viser bare noen av verdiene som blir lagret for en oppgave.

Show

Show klassen inneholder kode for å kunne gjøre filene assosiert med en oppgave tilgjengelige bare for de som skal ha tilgang til dem. Studenter som jobber med sin egen oppgave, hovedveileder, sensor og trykkeriet.

Denne klassen ligger ikke i *lib/* katalogen, men i *htdocs/* katalogen.

```
Class:    show.php
Include: auth.php
           daim.inc
           db.php
Methods:
katalog($id=null)
check_access()
download($name=null,$mimetype)
```

Template

Template klassen brukes for å laste inn en template og lagre verdiene inn i templatens før den genererte html koden vises på skjermen. Templates er hendige i forhold til gjenbruk av html kode.

Templatene som lastes inn og utfylles, ligger i *inc/* katalogen.

```
Class:    template.php
Include:
Methods:
__construct($name=null,$content=null,$dir=null)
setTemplate($name=null)
getTemplate()
setContent($content=null)
expand($content=null)
```

Update

Update klassen er klassen som inneholder kode for å *validere informasjon* og *skrive til databasen*. Klassen leser fra form variabler og legger informasjon fra disse inn i databasen dersom valideringen av informasjon godkjennes.

Denne klassen er den som oppdateres oftest, siden nye sider som skal lagre verdier må legge til en eller flere metoder for lagring i update klassen. Dersom det oppstår en kodefeil her kan all lagring slutt å fungere. Det er derfor veldig viktig å teste koden nøye både før og etter man inkluderer ny kode i denne filen.

Class: update.php

Include: auth.php
 daim.inc
 dates.php
 db.php
 kandidat.php
 oppgave.php

Methods:

```
bekreft_gruppe()  
frist_innlevering()  
beregneFrist($startdato=null,$studietype=null,$utland=null)  
endre_student()  
legg_inn_medstudenter()  
sjekk_medstudenter($bekreftet=null)  
legg_inn_veiledere()  
legg_inn_biblio()  
lagre_upload()  
lagre_hovedveil()  
lagre_sensor()  
lagre_sensur_honorar()  
sjekkKandidat()  
lagre_brukernavn()  
lagre_sensor_oppgave()  
lagre_oppgavetekst()  
lagre_oppgave_admin()  
sjekk_datoer($oppgave_id,$datoer,$dato_start,$dato_frist,&$msg)  
godkjenn_kontrakt(&$daimdb,$oppgave_id,$student_id,$status)  
fjern_fra_gruppe(&$daimdb,$oppgave_id,$student_id,$status)  
endre_status(&$daimdb,$oppgave_id,$student_id,$status)  
visStatus($msg,$t=null)  
oppgave_detalj_reload($student_id=null,$msg=null)  
dbDate($d=null)  
lagre_inst_info()  
lagre_studieprog()  
lagre_admin()
```

Include filer

Include filer ligger i *inc/* katalogen. Dette er ikke klassefiler, men filer som inneholder kode som blir gjenbrukt i flere sammenhenger. De forskjellige template filene ligger også i denne katalogen.

Daim

Daim er en include fil, og daim.inc filen er rammeverket for alle sidene i DAIM systemet. Filen inneholder metodene for å tegne opp nettsidene, og laste inn stilarkene for DAIM. Globale konstanter blir også definert i denne hovedfilen.

Studentsider genereres ved å gjøre bruk av *page_start()* og *page_end()*. Det som skal vises på siden skrives som HTML mellom de to funksjonene. Administrasjonssidene bruker *admin_page_start()* og *admin_page_end()* for å generere nettsider.

File: daim.inc

Constants:

```
$base_href=$_SERVER['DOCUMENT_ROOT'];
$browser=browser();
$server=$_SERVER['SERVER_NAME'];
define('READ_ONLY',1);
define('ORG',"Institutt for ... - NTNU");
define('BASE_HREF',"http://$server");
define('HOVED_TITTEL','DAIM - Masteroppgave');
define('MAIL_TO','daim-admin@idi.ntnu.no');
define('SMS_TO','91897703@mobilpost.no');

//FEILKODER
define('ERR_IKKE_MASTERSTUDENT',
"Beklager, du er ikke en registert masterstudent i databasen.\n".
"Er dette feil kontakt administrasjon ved ditt institutt. \n".
"Ta med studentbevis.");
define('ERR_AUTH',"Kan ikke opprette auth-objekt",LOG_ERR);
define('ERR_KANDIDAT',"Kan ikke opprette auth-objekt",LOG_ERR);
define('ERR_OPPGAVE',"Kan ikke opprette auth-objekt",LOG_ERR);

$doctype='<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd">';
```

```
$oppgave_dir="/srv/web/$server/oppgaver";
```

Functions:

```
page_start($page_title=null,$page_heading=null,$onload=null,
           $script=null,$style=null)
page_end($footer=ORG)
```



```

admin_page_start($page_title=null,$page_heading=null,$onload=null
                ,$script=null,$style=null)
admin_page_end($footer=ORG)
errmsg($msg,$loglevel=null,$url="/",$hovedtittel=null,
        $undertittel='Masteroppgave',$delay=null,
        $in_frame=null,$heading=null)
norskDato($date,$format=null,$use_time=null)
set_form_id($name=null,$reuse=null)
verify_form_id($debug=null)
send_browser_til($url,$delay=0)
quote($s=null)
addField($fnavn=null,$felt=null,$num=null,$use_null=true,
        $add_missing=null,$s=',')
print_v($var,$do_exit=1) // Min versjon av print_r...
validateStatus($K=null,$min_status=null,$max_status=null)
numericStatus($status=null)
browser()
isIE()
js_meny($obj,$id,$template,$meny=null,$parent='',
        $selected=null,&$fsize)
statusinfo($status='registert')
alert($msg=null,$do_exit=true)

```

Meny

Meny er include filen for menyen i administrasjonsgrensesnittet. Hvilke knapper som vises for administratorbrukeren er avhengig av hvilken rolle brukeren har for øyeblikket.

For å forhindre at menyen ble alt for stor for systemadministratorer ble det laget en egen oversiktside for funksjonaliteten systemadministratorer har.

File: meny.inc

Functions: <ingen>

Dynamikk i nettsidene

I administrasjonssidene for DAIM systemet er det mye dynamikk, og sidene er relativt komplekst oppbygd. Idealet har vært å lage nettsider som er dynamiske, brukervennlige og ikke belaster nett-tjeneren i for stor grad.

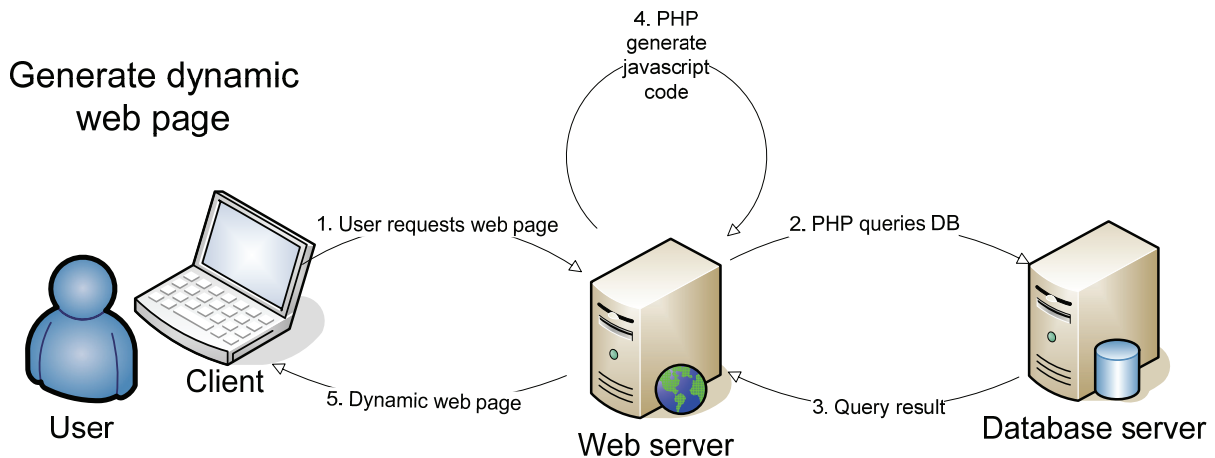
Teknologiene som benyttes for å lage dynamiske sider er: **PHP 5.0.3** og **MySQL 5** for dynamikk på tjeneren, og **Javascript** for dynamikk på klient siden.

Det er en del hovedprinsipp som går igjen i de dynamiske sidene:

1. Informasjonen lastes en gang fra nett-tjener.
2. Javascript for dynamikk i nettleseren.
3. Informasjon som lagres valideres nærmest mulig brukeren.

Informasjonen lastes en gang fra nett-tjener

For å minimere antall aksesser til tjeneren, lastes relevant informasjon hentet fra databasen inn i genererte lagringsstrukturer i Javascript, i det siden lastes første gang. PHP brukes for å generere Javascriptkode for informasjonsstrukturene som benyttes; dynamisk PHP kode genererer dynamisk Javascriptkode.



Dynamisk generering av nettside. Siden lastes *en* gang fra tjener til klient.

Javascript for dynamikk i nettleseren

I administrasjonssidene brukes menyer for å velge hvilken informasjon man ønsker å arbeide med. Javascript brukes til å håndtering av valg fra menyer. Valg gjøres i ei meny, og verdiene for valgt menyinnslag opplastes til editerbare felter.

PHP og SQL eksisterer på tjeneren mens Javascript finnes på klienten, og derfor finnes det ingen direkte kommunikasjon. At PHP genererer Javascriptkode er en løsning for å få dynamiske verdier på tjener overført til klient. Relevante verdier lastes fra databasen til klientens nettleser ved første aksess.

Å laste informasjon fra databasen inn i generert Javascriptkode gir:

- Færre aksesser mot nett-tjeneren enn ren tjenerbasert dynamikk.
- Sømløs brukeropplevelse; ingen venting på tjeneren for innlasting av ny verdier.

Ulemper er at informasjonen som opplastes til en nettside kan utdateres.

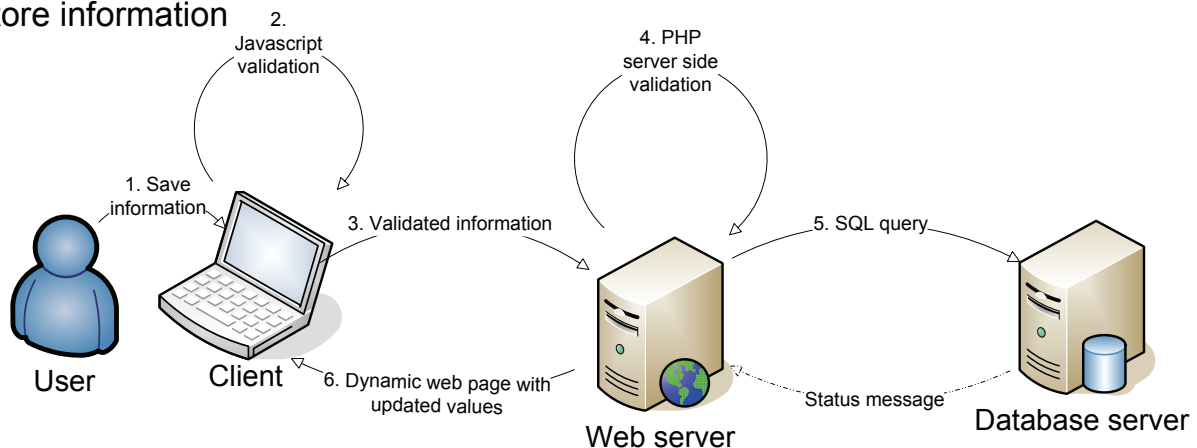
Sesjonsvariabler i DAIM har 20 minutters levetid. Etter det må informasjon hentes fra tjeneren på nytt.

Informasjon som lagres valideres nærmest mulig brukeren

Javascript er uegnet til kritisk validering av data. Grunnen til at Javascript er uegnet for validering på grunn av manglende kontroll klientens nettleser. Det er bare å slå av Javascript for å forhindre klient validering.

Hvorfor validere informasjon på klienten når man ikke kan stole på resultatet? Feil eller manglende informasjon bør gis beskjed om nærmest mulig brukeren. Tjeneren spares også for unødvendig aksess ved å stoppe mesteparten av feil i klienten.

Validate and
store information



Validering av informasjon ved lagring.

Informasjon som lagres bør alltid valideres på tjeneren, for kun der har man kontroll over at valideringen faktisk utføres. Valideringen i PHP forhindrer at en bruker kan slå av Javascript for å komme forbi valideringsrutiner. Normalt sett skal ikke feil oppdages på tjeneren dersom valideringen i Javascript ble kjørt. Korrupte verdier vil også stanses av validering på tjeneren.

Unntakene

Siden for å vise alle masteroppgavene i administrasjonsgrensesnittet laster inn for mange verdier til at det er hensiktsmessig å laste alle verdiene over til klienten ved første lasting av siden. Derfor benyttes det XMLHTTP på denne siden for å dynamisk kunne laste inn detaljer for valgt oppgave i det oppgaven velges fra menyen. Det er kun denne siden som benytter seg av XMLHTTP i DAIM systemet.

En annen måte å håndtere kommunikasjon mellom Javascript og PHP er via iFrames. iFrames er gjort bruk av i blant annet oversiktssidene over sensorer og hovedveiledere.