

Administrasjonsverktøy for Amnesty Jus Oslo

Gruppe 19

Margit Cecilie Haugen s163289
Tonje Henriksen s156049
Pernille Mohn s163300

30.05.2012



HØGSKOLEN I OSLO
OG AKERSHUS

Studieprogram: Informasjonsteknologi
Postadresse: Postboks 4 St. Olavs plass, 0130 Oslo
Besøksadresse: Holbergs plass, Oslo

PROSJEKT NR.

19

TILGJENGELIGHET

Åpen

Telefon: 22 45 32 00

Telefaks: 22 45 32 05

HOVEDPROSJEKT

HOVEDPROSJEKTETS TITTEL	DATO 30.05.2012
Administrasjonsverktøy for Amnesty Jus Oslo	ANTALL SIDER / BILAG 91
PROSJEKTDELTAKERE	INTERN VEILEDER
Tonje Henriksen – s156049 Margit Cecilie Haugen – s163289 Pernille Mohn – s163300	Torunn Gjester

OPPDAGSGIVER	KONTAKTPERSON
Amnesty Jus Oslo	Alf Butenschøn Skre

SAMMENDRAG

Dette prosjektet er utviklet for oppdragsgiveren Amnesty Jus Oslo, organisasjonen ønsket å få en bedre oversikt over sine medlemmer, og en bedre kommunikasjonsmåte med medlemmene. På oppdrag fra dem skulle vi komme frem til en god løsning for dette. Vi satt i startfasen av prosjektet sammen en kravspesifikasjon i samarbeid med styret, og kom frem til en løsning vi kunne utvikle og som tilfredsstilte ønskene Amnesty Jus hadde.

Det har blitt utviklet en medlemsside og administrasjonsside, samt en liten Android mobilapplikasjon, hvor styret kan få listet medlemmer og grupper, og sendt ut informasjons-SMS til medlemmer

Vi har over et semester utviklet en webløsning og en applikasjon for Amnesty Jus Oslo. Dette er dokumentasjonen for løsningene vi har kommet frem til. Det er delt inn i to deler, en prosjektrapport hvor utviklingen står dokumentert og en prosessrapport der prosjektet prosess er dokumentert.

3 STIKKORD
Medlemsregister

Administrasjonssystem

Android applikasjon

Forord

Administrasjonsverktøy for Amnesty Jus Oslo er et avslutningsprosjekt etter tre årlig utdannelse ved Høgskolen i Oslo og Akershus - Anvendt Dataeknologi, og har vart over vårsemesteret 2012.

Prosjektets deltagere er: Pernille Mohn, Tonje Henriksen, Margit C. Haugen.

Prosjektet er utført for arbeidsgiver: Amnesty Jus Oslo, en studentorganisasjon ved Universitet i Oslo juridisk fakultet.

Prosjektrapporten er dokumentasjon på administrasjonsverktøyet som vi har utviklet. Det er en medlemsside for registrerings- og administrasjonsside, og en Android mobilapplikasjon. Rapporten har blitt delt inn i to deler, prosjektrapport og prosessrapport. All kildekode vil ligge på hjemmesiden, samt sammen med dagbok og hele rapporten i .pdf format.

Android mobilapplikasjonen vil ligge på hjemmesiden med en .apk fil, som kan installeres på Android mobiltelefoner fra versjon 2.3.x og oppover, ellers vil kildekoden ligge der, så da kan det kjøres i en Android emulator.

Hjemmesiden: <http://www.stud.hio.no/~s156049/hovedprosjekt/>

Vi ønsker å takke:

- Vår veileder Torunn Gjester for bra oppfølging under hele prosjektperioden .
- Amnesty Jus som oppdragsgiver.

Signatur:

Sted/ Dato

Pernille Mohn

Tonje Henriksen

Margit C. Haugen

Innholdsfortegnelse

Produktrapport.....	5
Prosessrapport.....	78
Kilder.....	85
Ordliste	85
Vedlegg.....	87

HØGSKOLEN I OSLO OG AKERSHUS

Prosessrapport

Administrasjonsverktøy for Amnesty Jus Oslo

**Margit Cecilie Haugen - s163289
Tonje Henriksen - s156049
Pernille Mohn - s163300**

5/30/2012

Innhold

1.	Innledning	9
2.	Analyse.....	9
2.1	Bakgrunn.....	9
2.2	Mål.....	10
2.3	Kravspesifikasjon.....	10
2.4	Use Case modell.....	11
2.5	Detaljert Use Case Beskrivelse	12
2.6	ER-modell.....	14
2.7	Sekvensdiagram	15
2.8	Sidekart.....	16
3.	Implementeringsfasen og produktrapport.....	18
3.1	HTML og CSS.....	18
3.2	PHP	18
3.3	SQL.....	18
3.4	JavaScript	18
3.5	Krypterings algoritmen md5.....	19
3.6	Databasene	19
3.6.1	Medlem	20
3.6.2	Post.....	21
3.7	Brukersiden.....	22
3.7.1	Registrering	22
3.7.2	Login	26
3.7.3	Medlemssiden	28
3.8	Administrasjonsdelen	30
3.8.1	Logg inn	30
3.8.2	Administrasjonsforside.....	31
3.8.3	Medlemmer.....	32
3.8.4	Grupper	48
3.8.5	Epost til grupper	49
3.8.6	Statistikk.....	51
3.8.7	Endre passord.....	57
3.8.8	Logg ut.....	59
3.9	Testing av systemet	60

3.10 Endringer.....	61
4. Mobilapplikasjon.....	63
4.1 Android	63
4.1.2 Java	63
4.1.3 eXtensible Markup Language (XML)	63
4.1.4 SQLite	64
4.2 Applikasjonens oppbygning	64
4.2.1 Manifestfilen.....	64
4.2.2 Layout.....	64
.....	64
4.2.3 Values	65
4.2.4 R.java	65
4.2.5 Java kildekode.....	65
4.2.6 JavaScript Object Notation (JSON).....	66
4.3 Applikasjonen AIJSapp	66
4.3.1 Applikasjonskart.....	66
.....	66
4.3.2 Manifestet.....	67
4.3.3 Hovedaktiviteten	69
4.3.4 Menylinjen.....	69
4.3.5 Oppdatering av databasen.....	71
4.3.6 Lokal database.....	72
4.3.7 Meldingsutsendig	73
4.3.8 Alle medlemmer.....	74
.....	75
4.3.9 Velg gruppe.....	76
5. Konklusjon.....	77
6. Kilder	85
7. Ordliste	85
8. Vedlegg.....	87

1. Innledning

Amnesty Jus Oslo tok kontakt med Høgskolen i Oslo og Akershus fordi de ønsket å forbedre oversikten over sine medlemmer og måten de kommuniserte med medlemmene sine på. Vi fikk oppdraget om å utvikle et administrasjonssystem for dem. Løsningen vi kom frem til i samarbeid med styret i Amnesty Jus Oslo, ble en tre delt løsning, men en medlemsdel, administrasjonsdel og en mobilapplikasjon.

2. Analyse

2.1 Bakgrunn

Amnesty Jus Oslo er Universitet i Oslo Det juridiske fakultet sin egen Amnesty International nettverk gruppe, som tar for seg menneskerettighets problemer. Det er et nettverk som består av flere undergruppe etter hvilke interesse områder de jobber med. Det som er problemet til Amnesty Jus er at de til dags dato ikke har noe ordentlig registrerings system, som har gjort det vanskelig for Amnesty Jus å holde oversikt over sine medlemmer, og å få ut informasjon om aktiviteter og nyheter som de ønsker å informere medlemmene om. Amnesty Jus ønsker å få et oversiktig databasesystem som gjør det enkelt for studenter å melde seg inn i nettverket, og en administrasjonsside for styret, der de kan få oversikt over medlemmer, grupper og hvor det vil være enkelt å få sendt ut informasjon til brukerne.

Vi har selv valgt å utvide oppdraget etter samtale med Amnesty til å utvikle en administrator Android applikasjon. Vi kom med dette forslaget i planleggingsfasen og la det frem for Amnesty, noe de var veldig interessert i. Det er utviklet en Android mobilapplikasjon der styret kan få oversikt over medlemmene i nettverket, og sendt ut sms direkte fra mobilen.



Firgur: Amnetyjus logo

2.2 Mål

- 1) Amnesty vil få oversikt over antall medlemmer.
 - a) Kunne få oversikt over alle aktive medlemmer.
 - b) Kunne få oversikt over alle senior medlemmer, dvs. De som har gått ut fra Juridiske fakultet.
- 2) Ønsker at studenter som har gått ut fra det Juridiske fakultet blir lagret i Senior medlemmer slik at de ennå har mulighet til å være aktive i Amnesty jus.
- 3) Designet av systemet skal være enkelt og lett forståelig for alle.
 - a) Det skal være enkelt for medlemmer å registrere seg.
 - b) Det skal være enkelt for medlemmer å melde seg ut igjen av nettverket.
 - c) Kunden skal få en e-post hvor de må validere at de melder seg inn

2.3 Kravspesifikasjon

1. Medlemsside

- 1.1. Nye medlemmer skal kunne registrere selv.
 - 1.1.1. Input skal bli validert
 - 1.1.2. Må validere sin bruker via e-post
 - 1.1.3. Nye medlemmer skal motta en velkomst e-post etter fullført registrering
- 1.2. Medlemmer skal kunne logge inn
 - 1.2.1. Det skal være en link tillogg inn ved hovedsiden
 - 1.2.2. Medlemmer skal kunne endre opplysninger om seg selv
 - 1.2.3. Medlemmer skal kunne slette sin egen bruker
- 1.3. Elever som er ferdig gått ut av juridiske fakultet skal bli lagret i senior-medlemmer.
 - 1.3.1. Senior medlemmer skal ha like rettigheter som nåværende studenter.
- 1.4. Medlemmer skal kunne velge om de vil motta eller ikke motta gratis informasjons SMS fra styret

2. Administrasjonsside for styret

- 2.1. Kun styret skal ha tilgang til administrasjonssiden
 - 2.1.1. Det er et felles brukernavn og passord for administrasjonssiden
 - 2.1.2. Det skal være en link for å logge inn ved hovedsiden
- 2.2. Mulighet for å legge til, endre og fjerne medlemmer
- 2.3. Mulighet for å legge til og fjerne undergrupper
- 2.4. Mulighet for å se på medlemsliste
 - 2.4.1. Mulighet for sortering av medlemmer etter id, navn, registreringsdato.
 - 2.4.2. Mulighet til å velge en av gruppene eller undergruppene
 - 2.4.3. Mulighet for å få medlemslisten i Excel fil
 - 2.4.4. Mulighet til å skrive ut medlemsliste
- 2.5. Mulighet for å sende ut e-post til enkelte medlemmer, undergrupper eller alle medlemmer.
 - 2.5.1. Det skal være mulighet til å legge med vedlegg i e-posten

- 2.6. Mulighet for å sende ut SMS til enkeltmedlemmer, undergrupper eller alle medlemmer.
- 2.7. Skal ha mulighet til å få ut statistikk over medlemmene.

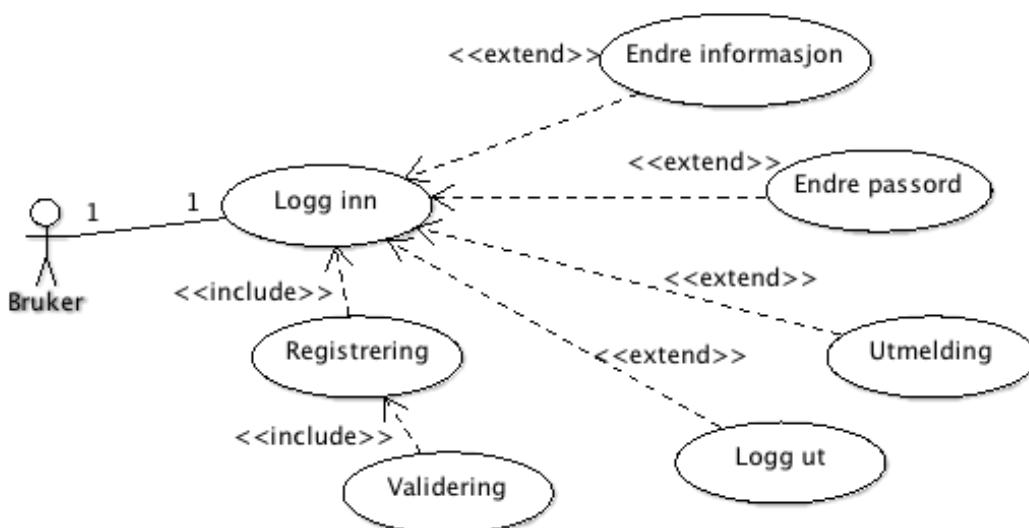
3. Android mobilapplikasjon for styret

- 3.1. Lage en fungerende mobilapplikasjon for Android
- 3.2. Applikasjonen skal ha kontakt med databasen Amnestyjus_no
- 3.3. Mulighet til å liste medlemmer
 - 3.3.1. All medlemsinformasjon om et enkelt medlem skal kunne vises.
 - 3.3.2. Oppdatert medlemsliste fra databasen.
- 3.4. Liste opp grupper i Amnesty Jus
 - 3.4.1. Oversikt over hvilke medlemmer som er i en gruppe.
 - 3.4.2. oppdatert gruppeliste fra databasen
- 3.5. Mulighet til å sende SMS til medlemmer

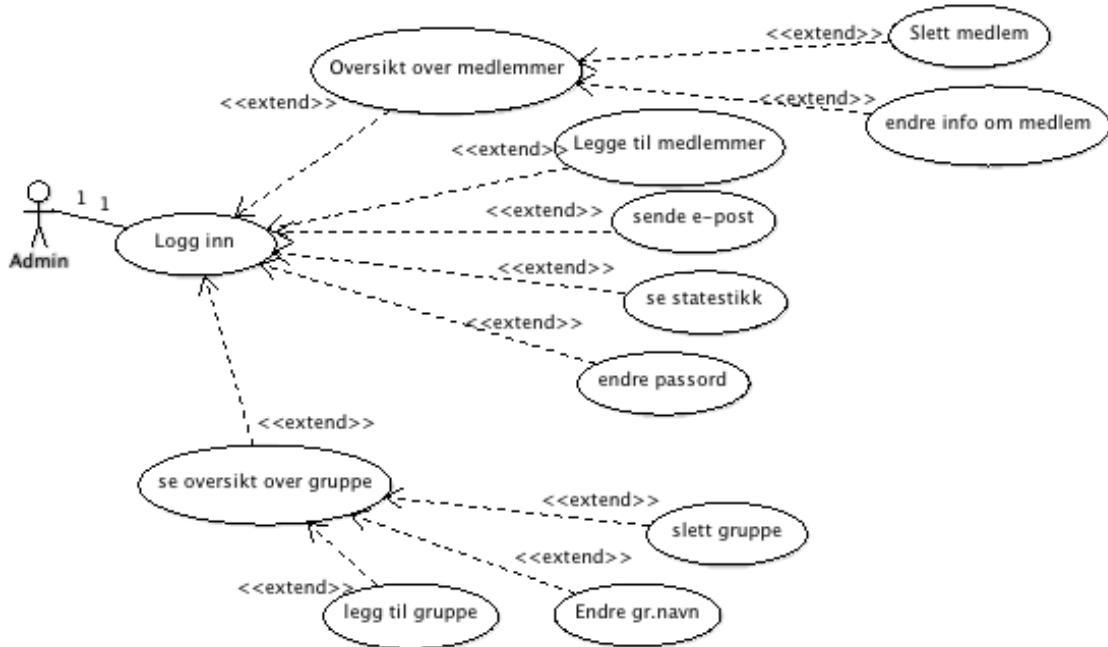
4. Tekniske krav

- 4.1. Programmerings språk: CSS, JavaScript, PHP, SQL, Java,
- 4.2. Markeringsspråk: HTML, XML
- 4.3. Datautveksling: JSON
- 4.4. Database: MySQL , SQLite

2.4 Use Case modell



Figur: Use Case modell for bruker



Figur: Use Case modell for administrator

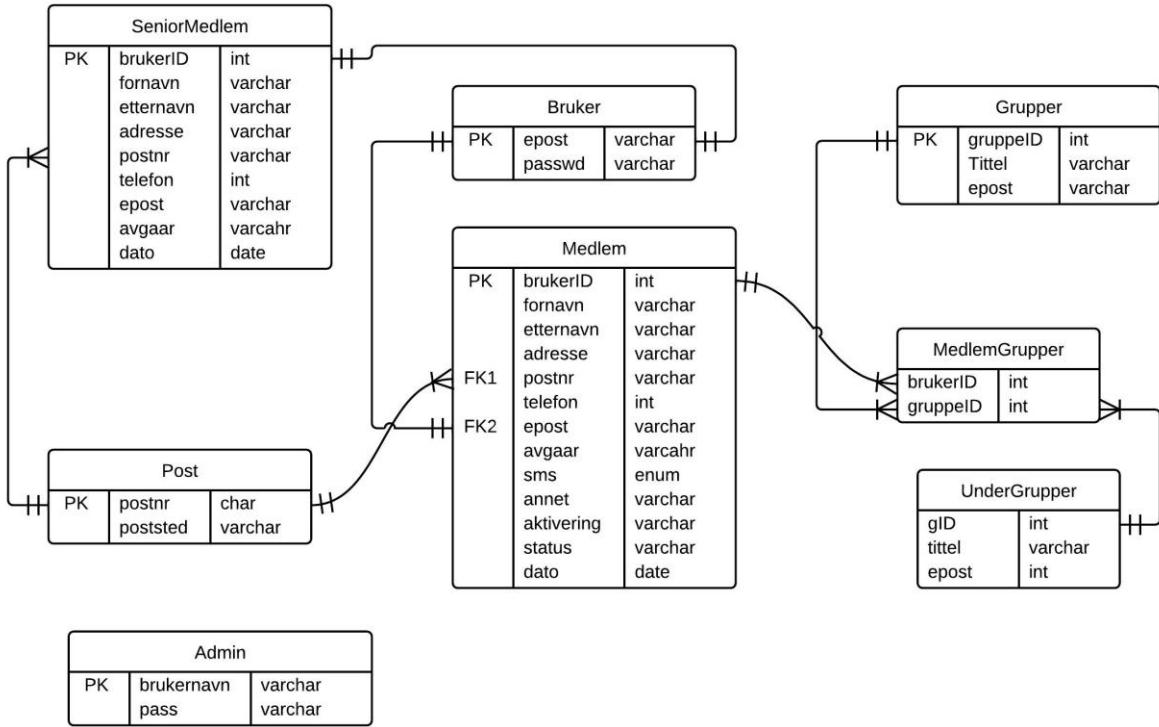
2.5 Detaljert Use Case Beskrivelse

Use case	Registrering
----------	--------------

Aktør	Nytt medlem/student
Pre betingelser	Student ønsker å bli medlem
Post betingelser	Student har blitt medlem
Hendelsesflyt	<ol style="list-style-type: none"> 1. Student fyller ut registreringsskjemaet. 2. Student trykker på knappen «Registrer». 3. Systemet sjekker om alle obligatoriske felt er fylt inn. 4. Systemet sender en validerings e-post til studenten. 5. Systemet legger inn brukerinformasjon i databasen. 6. Studenten validerer e-post og kontoen blir aktivert 7. Systemet sender en velkomst e-post til Studenten.
Variasjoner	<ol style="list-style-type: none"> 1) Studenten trykker på knappen «Tøm skjema». <ol style="list-style-type: none"> (a) Studenten må fylle ut alt på nytt 2) Alle obligatoriske felt er ikke fylt inn. <ol style="list-style-type: none"> (a) Systemet går tilbake til registreringsskjemaet. 3) Systemet får ikke lagret i databasen. <ol style="list-style-type: none"> (a) Systemet gir ut en feilmelding. 4) Bruker validerer ikke e-post. <ol style="list-style-type: none"> (a) bruker blir slettet når listen blir oppdatert. 6. Brukeren mottar ikke velkomstmeldingen.

Use case	Logg inn
Aktør	Medlem
Pre betingelser	Medlem ønsker å logge inn på siden.
Post betingelser	Medlem er logget på siden.
Hendelsesflyt	<ol style="list-style-type: none"> 1. Medlem skriver inn brukernavn og passord. 2. Medlem trykker på knappen «Logg inn». 3. Systemet sammenlikner brukernavn-passord kombinasjonen brukeren skrev inn med det samme som i databasen. 4. Systemet logger inn medlemmet
Variasjoner	<ol style="list-style-type: none"> 1) Medlemmet skriver inn feil brukernavn og passord. <ol style="list-style-type: none"> (a) Må skrive inn brukernavn og passord på nytt. 2) Medlemmet trykker på knappen «Avbryt» istedenfor. 3) Systemet får ikke kontakt med databasen. <ol style="list-style-type: none"> (a) Det kommer opp en feilmelding. (b) Medlemmet må prøve på nytt. 4) Medlem har glemt passord. <ol style="list-style-type: none"> (a) Må fylle inn gyldig e-post. (b) Får tilsendt nytt passord. (c) Må logge inn med det nye passordet. 5) Medlemmet er ikke registrert. <ol style="list-style-type: none"> (a) Studenten må melde seg inn.

2.6 ER-modell

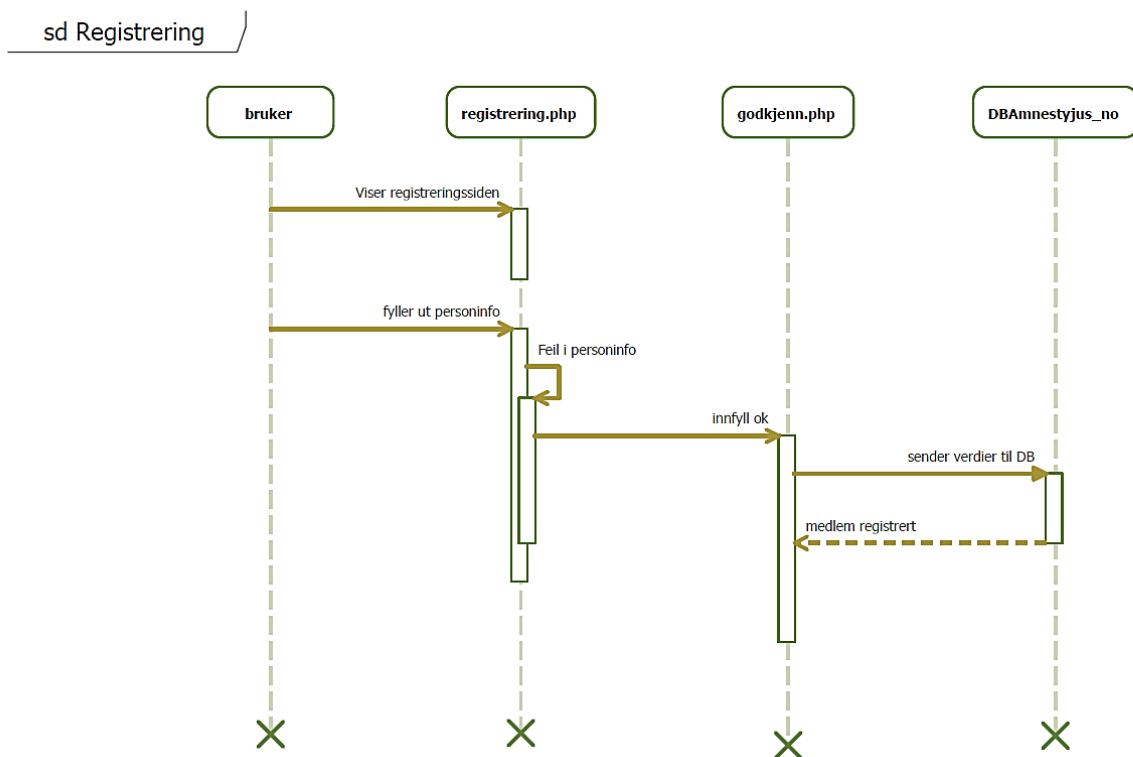


Figur: ER-modell over databasen.

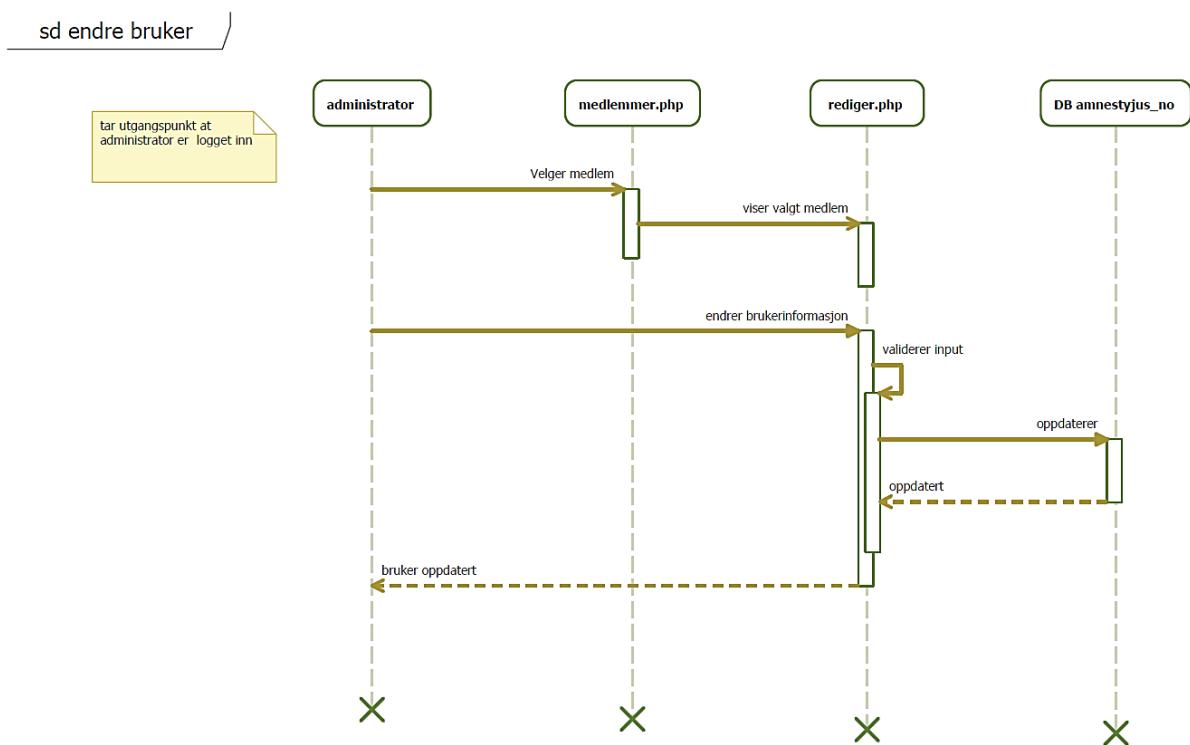
Dette er en ER-modell for databasesystemet til Amnesty Jus Oslo, med datatyper. Det er her vist med hvilke av verdiene som er primary key (PK) og hvem som er foreign key, og hvilke tabeller som henger sammen.

2.7 Sekvensdiagram

Sekvensdiagram er med for å vise hvordan systemet oppfører seg i noen hendelser.



Figur: Sekvensdiagram over registrering



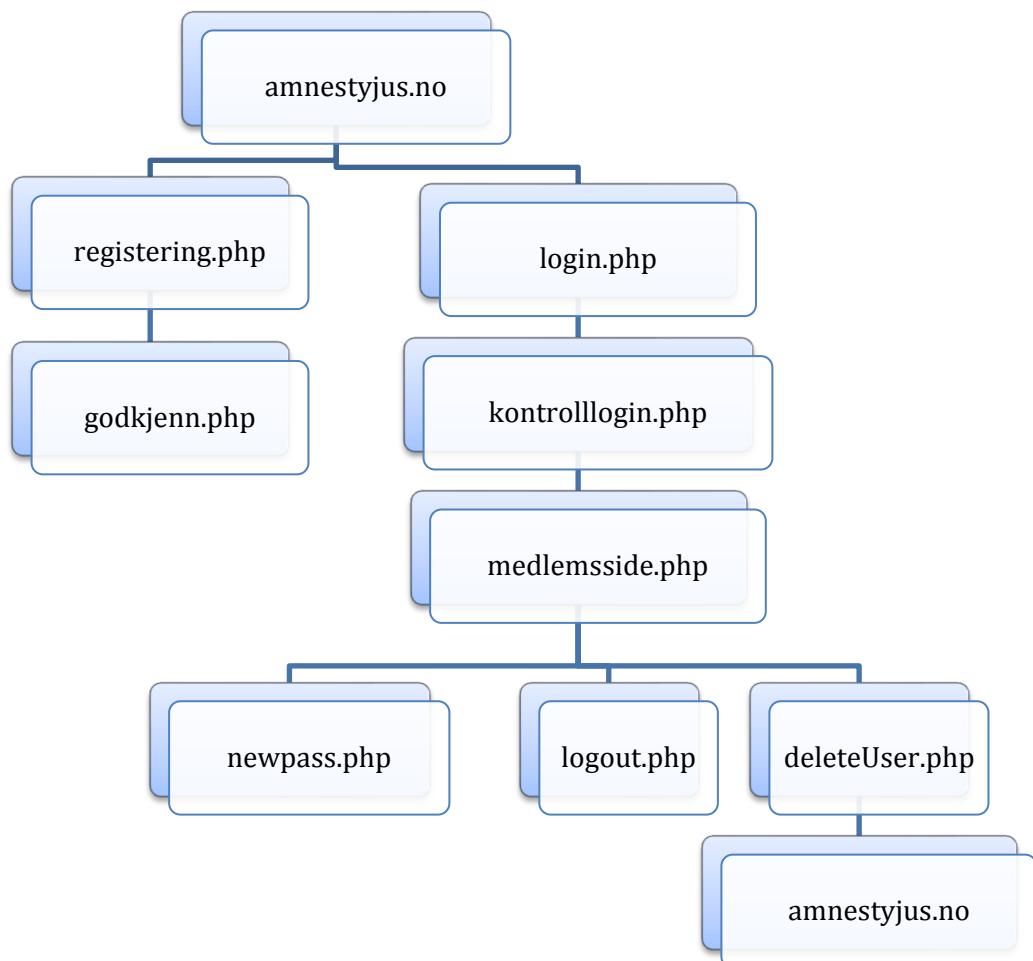
Figur: Sekvensdiagram over å endre bruker administrator.

2.8 Sidekart

Sidekart gir et bedre overblikk i hvordan sidene våre egentlig er bygget opp og hva som er tilgjengelig under hvert punkt.

Medlemssiden

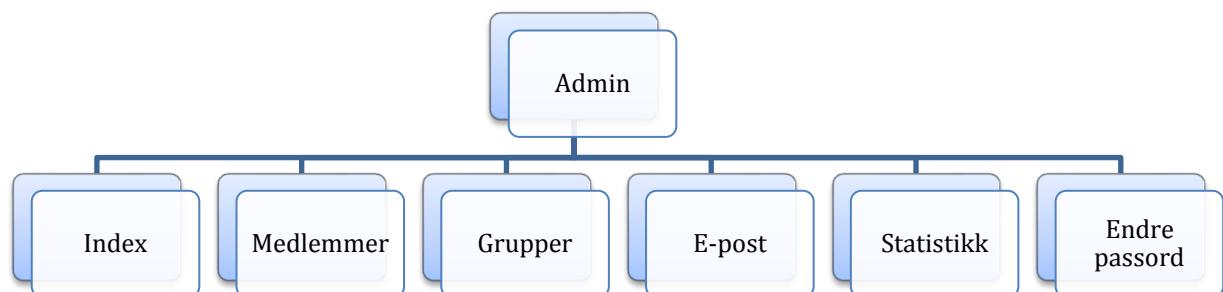
Dette sidekartet viser hvordan alle filene på medlemssiden henger sammen – med registrering på den ene siden oglogg inn på den andre siden. Noen av disse filene kjører i bakgrunnen og synes derfor ikke for brukerne.



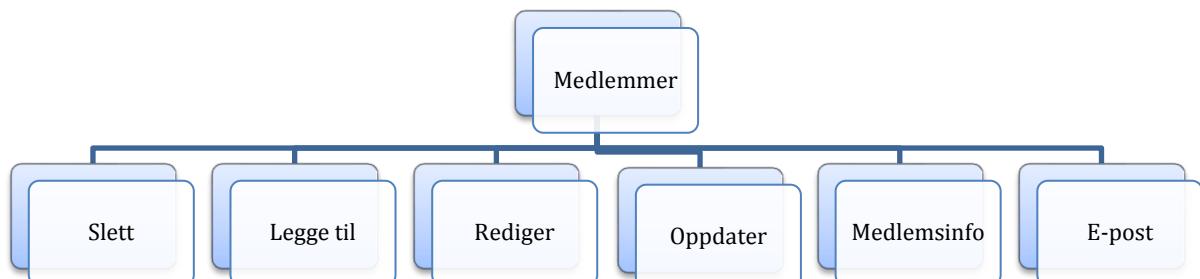
Figur: Sidekart over medlemssiden.

Administratorsiden

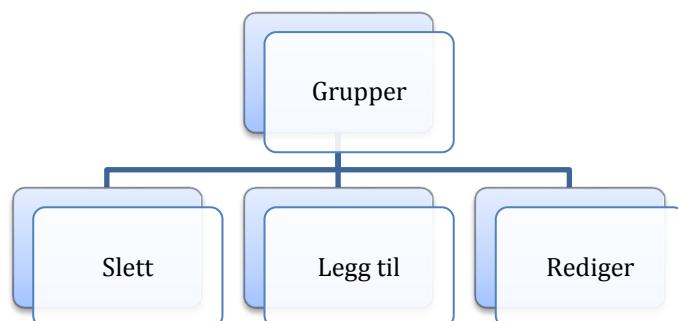
Disse sidekartene viser de seks kategoriene i hovedmenyen på administratorsiden med hvilke muligheter som kan velges under medlemmer.php og grupper.php.



Figur: Sidekart som viser hovedmenyen



Figur: Sidekart som viser mulighetene på medlemmer.php



Figur: Sidekart som viser mulighetene på grupper.php

3. Implementeringsfasen og produktrapport

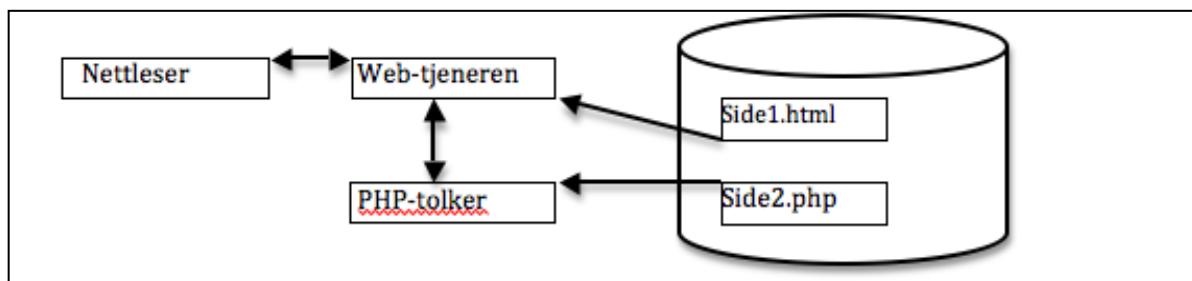
3.1 HTML og CSS

HTML står for Hyper Text Markup Language som er et markeringsspråk ikke et programmeringsspråk, et markeringsspråk vil si at den kombinerer tekst og ekstra informasjon for eksempel strukturen på teksten og oppsett. Det brukes til å formatere nettsider og er det mest kjente markeringsspråket. Den grammatiske strukturen er HTML DTD, som bruker SGML syntaks. SGML, Standard General Markup Language, spesifiserer et metaspråk som brukes til å angi oppbygningen av markeringsspråket.

Et HTML dokument består av elementer avgrenset av tagger med forskjellige betydning. Med den "taggede" informasjonen får man et html dokument som kan legges ut på nettside og leses av en nettleser. Siden HTML ikke har lagt høyde for det visuelle, brukes språket Cascading Style Sheet (CSS), det brukes til å definere utseende til en html fil. Definerer oppsett, farger og annen stilinformasjon, en sterk side ved CSS er at den kan brukes til å styre flere HTML filer, så hvis man for eksempel vil bytte bakgrunnsfarge på en nettside med flere html filer, endres dette bare i CSS filen.

3.2 PHP

PHP står for hypertext preprocessor, som er et skriftspråk som blir brukt til å utvikle web-applikasjoner. Et PHP-dokument blir lagret som en tekstfil på web-tjeneren og blir tolket og utført for hver forespørsel. Det er vanlig å kombinere vanlig HTML-kode og PHP-instruksjoner. PHP har også innebygget gode funksjoner for å koble en side opp i mot databasesystemer.



Figur. Utførelse av PHP-skript, Databasesystemer (Kristoffersen, 2009)

3.3 SQL

Til database arbeidet har vi brukt SQL språket, som står for Structured Query Language. SQL benyttes til å formulere og kjøre mot relasjonsdatabaser. Til dags dato er det mest brukte kontrollgrensesnittet til databaser SQL.

3.4 JavaScript

JavaScript er et skriftspråk som er mest brukt til å tilføre dynamisk elementer til en nettsiden. Det kan brukes både på tjener og klientsiden av en web-applikasjon. Det er sammen med HTML-koden (øvrig innhold på siden) JavaScript sender koden som skal

kjøres lokalt i nettleseren, som reaksjon av brukervalg på siden eller automatiske, også kalt en interaktiv webside.

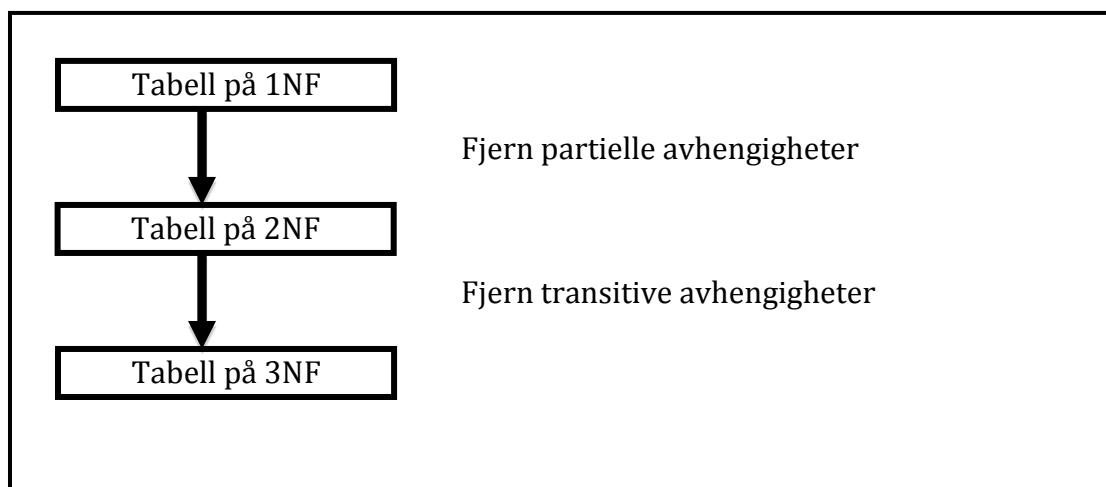
3.5 Krypterings algoritmen md5

Sikkerhet i et system er viktig så det ikke skjer uønskede hendelser. Vi har valgt å kun kryptere passordet i databasen siden den andre informasjon som blir lagret ikke er beregnet som sensitiv informasjon i forhold til personopplysningsloven § 2 punkt 8.

Vi har valgt å bruke md5 algoritmen til kryptering av passordet. Md5 Message-Digest Algorithm er en symmetrisk enveiskrypterings algoritme, dvs. kryptert data lar seg ikke dekryptere og en verdi vil alltid bli kryptert likt. Md5 lagrer en sjekksum av passordet som brukeren lagrer, når passordet blir oppgitt ved innlogging sjekkes passordet opp mot sjekksummen som er lagret, og ser at den er lik. Det positive med md5 er at passordet kan verifiseres uten at det er lagret i klartekst.

3.6 Databasene

Databasen er en relasjonsdatabase, det vil si at tabellene er forbundet med henvisninger eller nøkler mellom seg. En relasjonsdatabase kan deles inn i normalformer, det er 3 normalfordelinger 1NF, 2NF, og 3NF. Teknikken med normalfordeling er å minimere duplisering av informasjon. Når informasjon er lagret flere steder i en tabell, kan endringer føre til at databasen blir inkonsistent når noe endres. De tre normalformene bygger på hverandre, det er ikke mulig å ha NF2 hvis den ikke er NF1 osv.



Figur: Normaliseringstegene, Databasesystemer (Kristoffersen, 2009)

I databasen Amnesty_no bruker vi 8 av tabellene:

3.6.1Medlem

Dette er tabellen som lagrer all personinformasjonen om studentene som er medlemmer.

```
CREATE TABLE `medlem` (
  `brukerID` int(11) NOT NULL auto_increment,
  `Fornavn` varchar(45) NOT NULL,
  `Etternavn` varchar(45) NOT NULL,
  `Adresse` varchar(45) NOT NULL,
  `Postnr` varchar(4) NOT NULL,
  `Telefon` int(8) NOT NULL,
  `Epost` varchar(60) NOT NULL,
  `Avgaar` int(4) NOT NULL,
  `sms` enum('J','N') NOT NULL default 'J',
  `Annet` varchar(90) NOT NULL,
  `Aktivering` varchar(100) NOT NULL,
  `Status` varchar(20) NOT NULL,
  `dato` date NOT NULL,
  PRIMARY KEY (`brukerID`),
  KEY `fk_postmedlem` (`Postnr`),
  KEY `epost` (`Epost`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=272 ;
```

Datatypen Auto_increment har vi brukt til brukerID, den gjør at det settes en int id automatisk på alle medlemmene når de registrerer seg. I de fleste kolonnene har vi valgt varchar som betyr at det er en tekst streng som skal lagres, fordelen med å bruke varchar ovenfor datatypen char er selv om det skal være plass til 45 tegn i for eksempel navn og et medlem har et navn med kun 4 tegn opptar den ikke resten av den lagrete plassen, slik at i dette tilfellet ikke vil bli lagret 40 blanke tegn.

Enum datatypen har vi brukt til SMS fordi medlemmene skal kunne velge å reserve seg mot informasjons SMS fra styre hvis de ønsker det, her ser du at vi har satt 'J', ja, som standard for SMS hvis de ikke gjør det om til 'N' nei ved utfylling av registreringsskjema.

Primærnøkkelen i tabellen er brukerID, det betyr at det kun en gang forekommer en bestemt brukerID. En primærnøkkel skal aldri inneholde nullmerker eller repetisjoner. Primærnøkkelen hjelper deg å bedre datakvaliteten i databasen. Den gir beskjed til databasen om og alltid sjekke at brukerID blir fylt ut og at det ikke er et annet eksisterende medlem med samme brukerID.

Til alle kolonnene i tabellen har vi satt nøkkelordene NOT NULL, det vil forby nullmerking for de utvalgte kolonne, altså at ingen kolonne blir lagret tom.

På slutten av tabellen har vi lagt med `ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=272 ;` som tillater at det kan skrives med tegnsettet latin1, som inkluderer de norske tegnene 'æ', 'ø', 'å'.

Siden databasen inneholder flere tabeller er det flere av de som hører logisk sammen med et en-til-mange forhold, i denne databasen er det for eksempel mellom tabellene post og medlem, postnummer i medlem er en referanse til postnummer i post tabellen, en slik referanse gjøres med en fremmednøkkel, KEY som du ser over i tabellen.

3.6.2 Post

```
CREATE TABLE IF NOT EXISTS `post` (
  `postnr` char(4) character set latin1 NOT NULL,
  `poststed` varchar(30) collate utf8_danish_ci default NULL,
  PRIMARY KEY (`postnr`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_danish_ci;
```

Poststed og postnummer har vi delt inn i en egen tabell som skrevet om tidligere i henhold til normalfordeling.

I posttabellen er det allerede lagt inn alle postnumrene i Norge, så det automatisk blir lagt til poststedet når medlemmet skriver inn postnummeret.

-Resten av databasene kan sees som vedlegg på slutten av rapporten.

For å koble til databasen har vi filen db.php, alle PHP filene som inkluderer db.php har tilgang til databasen, hvis den ikke får kontakt vises en feilmelding.

```
<?php
$DB_knytning = mysql_connect('amnestyjus.no.mysql', 'amnestyjus_no', 'cdMRccWW');

if ($DB_knytning == false)
{
    die("Kunne ikke knytte meg til server".mysql_error());
}

if(!mysql_select_db('amnestyjus_no'))
{
    echo mysql_error();
}
mysql_query ('SET NAMES `utf8`');
?>
```

3.7Brukersiden.

3.7.1 Registrering

Amnesty Jus Oslo

AMNESTY INTERNATIONAL

Home Aktivitetskalender Gruppene Kontakt oss / Bli medlem Om Amnesty Jus Oslo

Personlig informasjon

E-post*	<input type="text"/>
Passord	<input type="password"/>
Bekreft passord	<input type="password"/>
Fornavn*	<input type="text"/>
Etternavn*	<input type="text"/>
Adresse*	<input type="text"/>
Postnummer:	<input type="text"/>

Ferske innlegg

- Hyttetur i Marka med Amnesty Jus Oslo!
- Kvinnedagen: Aksjon på Domus Nova
- De nominerte er...
- AJJS inviterer til GENERALFORSAMLING
- Infomøte om AJJS

Kommende hendelser

MAY	11	Fri	12:30 Lunsj på kontoret
MAY	18	Fri	12:30 Lunsj på kontoret
MAY	25	Fri	12:30 Lunsj på kontoret
JUN	1	Sa	12:30 Lunsj på kontoret

Figur: Bilde registrering.php

Registrering.php filen kan studentene ved juridisk fakultet registrere seg som medlemmer i amnestyjus, de må registrere seg med personlig informasjon som blir send til databasen Amnestyjus_no og registrert i medlemstabellen.

```

if ($_SESSION['fornavn']=='' || !preg_match('/^([A-Za-z\æøå]+)$/', $_SESSION['fornavn']))
    $arrErrors['fornavn'] = 'Vennligst skriv inn et fornavn';
if ($_SESSION['etternavn']=='' || !preg_match('/^([A-Za-z\æøå]+)$/', $_SESSION['etternavn']))
    $arrErrors['etternavn'] = 'Vennligst skriv inn et etternavn';
if (!preg_match('/^([a-z0-9-]+([a-z0-9-]+)*@[a-z0-9-]+(\.[a-z]{2,3})+$/', $_SESSION['epost'])){
    $arrErrors['epost'] = 'Vennligst skriv inn en korrekt epostadresse';
}
if ($_SESSION['tlf']=='' || !preg_match('/^([0-9]+)$/', $_SESSION['tlf']))
    $arrErrors['tlf'] = 'Vennligst skriv inn et gyldig telefonnummer';

if($_SESSION['pasw'] == $_SESSION['paswok']){
    if ($_SESSION['pasw']=='' || !preg_match("/^(?=.*[0-9])(?=.*[a-zA-Z]).{8,}$/", $_SESSION['pasw']))
        $arrErrors['pasw'] = 'Vennligst skriv inn ett passord som inneholder 8, små og store bokstaver og tall.';
}
else{
    $arrErrors['paswok'] = 'Dine passord er ikke like.';
}

if (count($arrErrors) == 0) {
    header("Location: godkjenn.php");
}

```

All informasjon studenten fyller inn blir validert ved hjelp av flere if-setninger. Dersom en feilsituasjon oppstår i forhold til de reglene vi har satt, vil en feilmelding lagres i \$arrErrors matrisen. På slutten av if-setning kontrolleres det om \$arrErrors == 0, da vil brukeren bli sendt videre til godkjenn.php, hvis \$arrError != 0 vil feilmeldingen vises der feilsituasjonen har oppstått i registrering.php filen, og brukeren må rette feilene i forhold til retningslinjene som blir oppgitt.

Personlig informasjon

E-post*	<input type="text" value="1234"/>	Vennligst skriv inn en korrekt epostadresse
Passord	<input type="text" value="*****"/>	Vennligst skriv inn ett passord som inneholder 8, små og store bokstaver og tall.
Bekreft passord	<input type="text" value="*****"/>	
Fornavn*	<input type="text"/>	Vennligst skriv inn et fornavn
Etternavn*	<input type="text"/>	Vennligst skriv inn et etternavn
Adresse*	<input type="text" value="veivei"/>	
Postnummer:	<input type="text" value="1167"/>	
	<input type="text" value="Oslo"/>	
Telefon*	<input type="text"/>	Vennligst skriv inn et gyldig telefonnummer

Figur: Bilde feilmelding ved registrering

E-posten til studenten vil bli lagret som brukernavn for brukerkontoen deres, e-posten vil bli sendt av filen godkjenn.php til databasen. I godkjenn.php blir variabelen \$sjekkEpost sendt ved hjelp av php for å kontrollere at e-post adressen ikke er registrert i databasen hos en annen bruker.

```
//sjekker om epost er i DB
$sjekkEpost ="SELECT Epost FROM medlem WHERE Epost = '$_SESSION[epost]'";
$resultat = mysql_query($sjekkEpost);
if(!$resultat){
    die(mysql_errno()." : ".mysql_error());
}
```

Studenten må oppgi et passord med tall, store og små bokstaver på minimum 8 tegn, de må bekrefte passordet en gang for å være sikker på at de har skrevet det passordet de ønsket å lagre riktig. Det vil blir sjekket med en if-setning i registrering.php som kontrollerer om passordene er like.

```
if($_SESSION['pasw'] == $_SESSION['paswok']){
    if ($_SESSION['pasw']=='' || !preg_match("/^.*(?:.{8,})(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).*$/", $_SESSION['pasw']))
        $arrErrors['pasw'] = 'Vennligst skriv inn ett passord som inneholder 8, små og store bokstaver og tall.';
}
else{
    $arrErrors['paswok'] = 'Dine passord er ikke like.';
}
```

Passordet vil bli lagret kryptert med krypterings algoritmen md5 i databasen, se punkt 3.5 om md5 algoritmen for bedre forklaring.

```
$pasw = md5($_SESSION['pasw']);
```

For at styret skal klare å ha oversikt over hvilke medlemmer som fortsatt er studenter ved juridiske fakultet ber de personen som ønsker å bli medlem skrive inn forventet avgangssår, studenten blir ikke slettet etter forventet avgangssår, men lagt i gruppen senior medlemmer i databasen, slik at hvis de ønsker å være aktive etter studie tiden er det fortsatt mulig.

Amnesty jus har mulighet til å sende ut SMS fra applikasjonen, dette er en gratis tjeneste for medlemmene, men de har rett til å reservere seg for dette hvis de ønsker det.

For at styret skal kunne fordele interesseområdene til medlemmer er det dannet noen undergrupper som de ber bruker krysse av ved registrering, bruker har mulighet til å endre hvilke grupper de ønsker å være medlem i ved å logge inn ved senere anledning. Gruppene blir forandret ettersom hvilke interesseområder studentene som er medlem har.

Når studenten som ønsker å bli medlem har registrert all den obligatoriske informasjonen, vil det bli sendt en validerings e-post til e-posten som studenten oppga

som sin e-post adresse. For å bli lagt til i bruker databasen som medlem må brukeren godkjenne valideringen ved å trykke på linken i e-posten.

```
echo "En epost har blitt sendt til \"._SESSION['epost']." med aktiveringsnøkkelen.  
Vennligst sjekk din epost for å fullføre registreringen";  
$to = $_SESSION['epost'];  
$subject = " Registrering Amnesty Jus Oslo";  
$message = " Velkommen som medlem i Amnesty International Juridisk Studentnettverk\r\n  
Du eller noen andre har brukt din epostadresse for å registrere ett nytt medlem i Amnesty Jus Oslo,  
for å fullføre registreringen vennligst følg denne linken\r\nhttp://www.amnestyjus.no/godkjenn.php?aktivering\r\n\r\n  
Vennlig hilsen Amnesty International Juridisk Studentnettverk\r http://www.amnestyjus.no";  
$headers = 'From: noreply@amnestyjus.no' . "\r\n" .  
          'Reply-To: medlem@amnestyjus.no' . "\r\n" .  
          'X-Mailer: PHP/' . phpversion();  
mail($to, $subject, $message, $headers);
```

I databasen amnestyjus_no vil studenten stå med status som verify inntil personen har godkjent validerings e-posten. Studenten har ikke mulighet til å logge seg inn på medlemssiden før status i databasen er done, som viser at studenten har godkjent e-posten.

Dette etter ønske fra Amnesty jus, fordi de ville ha en sikkerhet i at ikke studenter meldte seg inn med andres eller ugyldige e-post adresser.

Done	aktivert	2012-05-07
195426149935956446818419167001849532428347366846	verify	2012-05-28

Figur: Bilde av aktivert og ikke aktivert i databasen.

3.7.2 Login

Amnesty Jus Oslo

Logg inn

Brukernavn/Epost	<input type="text"/>
Passord	<input type="password"/>
<input type="button" value="Login"/>	

[Vil du registrere deg?](#)

Ferske innlegg

- Hyttetur i Marka med Amnesty Jus Oslo!
- Kvinnedagen: Aksjon på Domus Nova
- De nominerte er...
- AIJS inviterer til GENERALFORSAMLING
- Infomøte om AIJS

Kommende hendelser

MAY 11 Fri	■ 12:30 Lunsj på kontoret
MAY 18 Fri	■ 12:30 Lunsj på kontoret
MAY 25 Fri	■ 12:30 Lunsj på kontoret
JUN 1	■ 12:30 Lunsj på kontoret

Figur: login.php

Ønsker et medlem å logge inn for å redigere noe av bruker informasjonen som er registrert i databasen kan medlemmet logge inn med brukernavnet og passordet som de registrerte når de meldte seg inn i Amnestyjus.

Har medlemmet glemt sitt passord har den mulighet til å få tilsendt et passord til e-post adressen.

Med funksjonen `createRandomPassword` blir det laget et tilfeldig passord, `$chars` er en string vi lager for å vise de bokstavene som kan brukes i passordet, vi setter at passordet skal være `<= 7` tegn, og med bokstaver og tall.

Det nye passordet vil så bli sendt til medlemmet på e-posten som er registrert, skriver bruker inne en e-post som ikke er lagret i databasen, vil den få beskjed om at ingen bruker er registrert med den e-posten, og få en link hvis studenten ønsker å registrere seg.

```

<?php
    if(isset($_REQUEST['nyttpass'])){
        $epost = ren($_REQUEST['epost']);
        $sjekkEpost ="SELECT Epost FROM medlem WHERE Epost = '$epost'";
        $resultat = mysql_query($sjekkEpost);
        if(!$resultat){
            die(mysql_errno()." : ".mysql_error());
        }
        if(mysql_num_rows($resultat) < 1){
            echo "Det er ikke en registrert med denne epostadressen, sjekk epostadressen,  

                eller <a href='registrering.php'>Registrer deg</a>";
        }
        else{
            $password = createRandomPassword();
            $passwd = md5($password);
            $sql =mysql_query("UPDATE bruker SET passwd='$passwd' WHERE Epost ='$epost'");
            if(!$sql){
                die(mysql_errno()." : ".mysql_error());
            }
            $epost = $epost;
            $subject = "Nytt passord";
            $message = "Ditt nye passord for medlemssidene på www.amnestyjus.no er: "
                ."$passwd.\n For å bevare sikkerheten på ditt passord,  

                burde du logge inn og endre ditt passord snarest mulig. \n\n
                Vennlig hilsen \nAmnesty Jus Oslo";
            $headers = 'From: noreply@amnestyjus.no' . "\r\n".
                'Reply-To: noreply@amnestyjus.no' . "\r\n".
                'X-Mailer: PHP/' . phpversion();
            mail($epost, $subject, $message, $headers);
            echo "Det er nå sendt et nytt passord til ".$epost;
        }
    }
    else{
        ?>
    }
}

```

I skriptet ser vi at det nye passordet vil erstatte det gamle passordet og bli lagret kryptert med algoritmen md5 i databasen. Siden det ikke er sikkert å ha et passord skrevet på e-post, vil medlemmet få informasjon om at det anbefales å endre passordet sitt snarest etter å ha mottatt e-posten.

Når medlemmet har logget inn med riktig brukernavn og passord, blir den sendt til medlemssiden.php.

Hvis en student som ikke har meldt seg inne kommer inn på login.php, er det lagt med en synlig link til registrering.php filen, som gjør det enkelt for studenten hvis den ønsker å bli medlem.

```

function createRandomPassword() {

    $chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    srand((double)microtime()*1000000);
    $i = 0;
    $pass = '' ;

    while ($i <= 7) {
        $num = rand() % 33;
        $tmp = substr($chars, $num, 1);
        $pass = $pass . $tmp;
        $i++;
    }

    return $pass;
}

```

3.7.3Medlemssiden

Home Aktivitetskalender Gruppene Kontakt oss / Bil medlem Om Amnesty Jus Oslo

Informasjon

Du er nå logget inn som Margit Haugen

E-post	haugen.margit@gmail.com
Fornavn	Margit
Etternavn	Haugen
Adresse	Breiblakkvn 14 j
Postnummer:	1167
Poststed:	
Telefon	92264344
Avgangssår	2012
Ønsker å motta SMS	<input type="checkbox"/>

- Seminargruppa
 Menneskerettighetsuka
 Faggruppen
 Kulturgruppen
 PR-gruppen
 Magasinsgruppen
 Komiteen

Logget inn som Margit

Medlemssiden

Endre passord

Utmelding

Logg ut

Ferske innlegg

- Hyttetur i Marka med Amnesty Jus Oslo!
- Kvinnedagen: Aksjon på Domus Nova
- De nominerte er...
- ALJS inviterer til GENERALFORSAMLING
- Infomøte om ALJS

Kommende hendelser

MAY 11 Fri	12:30 Lunsj på kontoret
MAY 18 Fri	12:30 Lunsj på kontoret
MAY 25 Fri	12:30 Lunsj på kontoret
JUN 1 Fri	12:30 Lunsj på kontoret
JUN 8 Fri	12:30 Lunsj på kontoret
JUN 15 Fri	12:30 Lunsj på kontoret
JUN 22 Fri	12:30 Lunsj på kontoret

Figur: medlemssiden.php

Redigering av brukerinformasjon kan medlemmet gjøre i filen medlemsside.php, det forutsett at medlemmet er innlogget. Der har medlemmet mulighet til å redigere all informasjonen som er lagret i dens rad i persondatabasen. Ønskes det å endre e-post som og er lagret som brukernavnet, vil det på nytt bli kontrollert med variabelen \$sjekkEpost at ikke e-postadressen allerede er registrert hos noen andre brukere. Etter at medlemmet har redigert den informasjonen som er ønsket, avsluttes prosessen ved å trykke på oppdater, da vil den nye informasjonen bli sendt til databasen Amnesty_no.

Ved å trykke på endre passord linken har brukeren mulighet til å endre det nåværende passordet i databasen. Medlemmet må da skrive inn det nåværende passord og det nye passordet. I det nye passordet som det tidligere passordet må det være tall, store og små bokstaver, og minimum 8 tegn. Passordet vil da bli lagret kryptert med md5 algoritmen i databasen.

```

if(isset($_POST['submit'])){
    $arrErrors = array();
    $passcheck = md5($_REQUEST['pass']);
    if($passcheck == $_SESSION['passcheck']){
        if($_REQUEST['newpass'] == $_REQUEST['newpass1']){
            if ($_REQUEST['newpass'] == '' || !preg_match("/^.*(?=.{8,})(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).*$/", $_REQUEST['newpass'])) {
                $arrErrors['newpass'] = 'Vennligst skriv inn ett passord som inneholder 8, små og store bokstaver og tall.';
            }
            else {
                $arrErrors['newpass2'] = 'Dine passord er ikke like.';
            }
        }
        else {
            $arrErrors['pass'] = 'Gammelt passord er feil';
        }
        if (count($arrErrors) == 0) {
            $newPass = md5($_REQUEST['newpass']);
            $result = mysql_query("UPDATE bruker SET passwd = '$newPass' WHERE epost = '$_SESSION[bruker]'")
                or die(mysql_error());
            $ok = "Ditt passord er nå oppdatert";
        }
    }
}

```

Hvis medlemmet skulle ønske å melde seg ut, har den mulighet til det ved å trykke på linken utmelding. Hvis medlemmet trykker på «Slett meg» knappen sjekkes det om medlemmet er sikker med å bruke JavaScript (beskrives nærmere under punkt 3.8.3). Da vil all bruker informasjon som var lagret om studenten bli fjernet fra databasen.

3.8 Administrasjonsdelen

3.8.1 Logg inn

The form is titled "ADMIN LOGIN". It contains two input fields: "Brukernavn:" and "Passord:". Below these is a checkbox labeled "Husk meg". At the bottom is a large blue button labeled "Logg inn".

Figur: login.php

Koden for å logge inn på administrasjon er nesten helt lik som koden for å logge inn medlemmer. Det eneste som er forandret er at brukernavn og passord hentes ut fra en annen tabell, og det faktum at vi har med en «husk meg» funksjon. Blir «husk meg» huket av settes det en cookie med brukernavnet. Denne cookien er gyldig i ett år, hvis den ikke er huket av settes det en session på brukernavnet.

```
$husk=$_POST['husk'];

$sql="SELECT * FROM admin WHERE brukernavn = '$brukernavn' and pass = '$passwd'";
$result=mysql_query($sql);
$count=mysql_num_rows($result); Tree

if($count==1) {
    if($husk=="on"){
        setcookie("brukernavn", $brukernavn, time()+60*60*24*100);
    }
    else if($husk==""){
        $_SESSION['brukernavn']= $brukernavn;
        $_SESSION['passwd'] = md5($passwd);
    }
    header("Location: index.php");
    exit();
}
?>
```

3.8.2 Administrasjonsforside

Øverst på hver side sjekkes det om brukeren er logget inn - om cookie eller session er satt. Hvis ikke cookie eller session er satt får ikke brukeren tilgang til siden, men heller sendt til login.php

```
if(! isset($_COOKIE['brukernavn'])){  
    if(! isset($_SESSION['brukernavn'])){  
        header("Location: login.php");  
    }  
}
```

Når man er logget inn vises index.php som første side. Dette er en helt enkel side hvor vi har valgt å legge inn beskrivende bilder og tekst for de mest brukte sidene. Dette har vi valgt slik det blir lettere tilgjengelig for brukertegn.



Figur: Bilde index.php

3.8.3 Medlemmer

Medlemmer.php gir en oversikt over alle registrerte medlemmer. Her er det en nedtrekksmeny hvor admin kan velge hvilken gruppe som skal vises. Gruppene som vises her hentes ut fra databasen i tilfelle det blir lagt til/slettet gruppe eller navnet blir endret på. I tillegg er det lagt til et ekstra felt hvor alle medlemmer vises uansett hvilken gruppe de er medlem i.

Figur: Søk medlemmer administrasjonssiden

Gruppene er delt inn i grupper og undergrupper og er derfor i to forskjellige tabeller. Det er brukt to forskjellige SQL spørninger, en for hovedgruppene og en for undergruppene, som hentes ut to while-løkker. Her legges gruppeID/gID inn som value, mens det er gruppenavnene som vises. I tillegg er feltene "Alle medlemmer" og "Seniormedlemmer" lagt inn manuelt. JavaScript funksjonen `onchange=«this.form.submit()»` er også lagt inn. Denne er lagt inn slik at vi ikke behøver å ha en submit knapp som må trykkes på for hver gang vi vil endre gruppevisning. Hvis JavaScript er skrudd av i den browseren som brukes vises en submit knapp. På denne måten vil nedtrekksmenyen fungere uavhengig av om JavaScript fungerer eller ikke.

```

<div class="index">
<fieldset style="float:left; margin-left:80px; width:200px;">
<form action="" method="post" name="gruppe">
<select name="q" onchange="this.form.submit()">
<option value='velg'>----- velg gruppe -----</option>
<option value='alle'>Alle medlemmer</option>
<option value='senior'>Seniormedlemmer</option>

<?php
while ($row = mysql_fetch_assoc($query)) {
echo "<option value=\"$row[gruppeID]\">$row[Tittel]</option>\n";
}
while ($row = mysql_fetch_assoc($query2)) {
echo "<option value=\"$row[gID]\">$row[Tittel]</option>\n";
}
?>
</select>
<noscript><br /><input type="submit" value="Velg" /></noscript>
</form>

```

Til høyre har vi et søkefelt hvor admin kan søke etter medlemmer. I koden sjekkes det først om søkerknappen er trykket, hvis den er det kjøres scriptet. Vi definerer variabelen \$search til å være innholdet i søkerfeltet, så hvis det er skrevet noe i søkerfeltet teller vi antall medlemmer, legger dette i en variabel, teller antall sider og legger dette i en variabel og henter ut medlemmene som matcher søkerkriteriet. Vi bruker jokertegnet % rundt variabelen \$search slik at de ikke trenger å angi nøyaktig hva de søker etter. Hvis det oppstår en feil i sql spørringen vises feilen og hvis søkereturer i 0 medlemmer vises meldingen «Ingen medlemmer». Hvis det ikke oppstår noen feil vises resultatet på samme måte som i gruppeoversikten.

```

if (isset ($_POST['sok'])) {
    $search=$_POST['search'];
    if ($search) {
        $adjacents = 3;
        $query = "SELECT COUNT( medlem.brukerID ) AS num
FROM medlem WHERE brukerID LIKE '%$search%' || Fornavn LIKE '%$search%' || Etternavn LIKE '%$search%'
|| Adresse LIKE '%$search%' || Postnr LIKE '%$search%' || Telefon LIKE '%$search%'
|| Epost LIKE '%$search%' || Avgaar LIKE '%$search%'";
        $total_pages = mysql_fetch_assoc(mysql_query($query));
        $total_pages = $total_pages[num];
        $sql = "SELECT * FROM medlem WHERE brukerID LIKE '%$search%' || Fornavn LIKE '%$search%' ||
Etternavn LIKE '%$search%' || Adresse LIKE '%$search%' || Postnr LIKE '%$search%' ||
Telefon LIKE '%$search%' || Epost LIKE '%$search%' ||
Avgaar LIKE '%$search%' ORDER BY medlem.brukertID ASC";
        $result = mysql_query($sql);
        if (!$result) {
            echo "Feil i spørringen ($sql): " . mysql_error();
            exit;
        }
        if (mysql_num_rows($result) == 0) {
            echo "<div class='center'>Ingen medlemmer </div><div class='spacer'></div>";
            exit;
        }
    }
}

```

Når «alle medlemmer» velges i nedtrekksmenyen vises det en tabell med alle medlemmene i den gruppen. Det vises ti medlemmer på hver side og hvis gruppen har flere medlemmer kommer det opp sidetall.

----- velg gruppe -----

Søk

Alle medlemmer

ID	Fornavn	Etternavn	Telefon	E-post	Avgangssår	sms	Dato	Valg	X
118	Tonje	haugen	93866085	jdidj@joe.no	2017	N	2012-01-12		
156	Tonje	Henriksen	456845	bestizznr1@hotmail.com	2014	J	2012-04-09		
156	Tonje	Henriksen	456845	bestizznr1@hotmail.com	2014	J	2012-04-09		
157	Tonje	Henriksen	7736263	bestizznr1@hotmail.com	2012	J	2012-04-09		
172	Tonje	Henriksen	456845	nklor@jope.jo	2016	J	2012-04-10		
174	Tonje	Henriksen	89020983	bestizznr1@hotmail.com	2012	J	2012-04-10		
175	Tonje	Henriksen	98626393	tonjehen@gmail.com	2012	J	2012-04-10		
175	Tonje	Henriksen	98626393	tonjehen@gmail.com	2012	J	2012-04-10		
176	Tonje	Henriksen	7736263	nklor@jope.jo	2014	J	2012-04-10		
202	Tonje	Henriksen	7896756	bedjopd@hotmail.com	2012	J	2011-02-13		

Antall medlemmer: 23

[Slett](#)
[Legg til medlem](#)
[1](#)
[2](#)
[3](#)
[neste >>](#)

[Oppdater](#)

Figur: søk av medlemmer

Trykker administrator på en av radene kommer man til medleminfo.php, hvor all informasjon som er lagret om dette medlemmet kommer opp. Vi har valgt bare å ha den viktigste informasjonen på medlemmer.php i tabellen og heller ha en annen side for å vise all informasjon slik at tabellen ikke blir for lang og uoversiktlig. For å kunne trykke på raden har vi valgt å bruke JavaScript og ikke en vanlig html link slik at man kan trykke på hele raden og ikke kun på teksten.

For å bruke javascript til dette har vi i head definert en funksjon som sier den skal gå til url 'en som blir satt når funksjonen kalles:

```
function DoNav(theUrl)
{
document.location.href = theUrl;
}
```

I tabellen bruker vi JavaScript funksjonen onclick og kaller på funksjonen i de alle kolonnene utenom den første og den siste. Url 'en som settes er medleminfo.php pluss brukerID på det medlemmet som trykkes på.

```

echo "<tr>";
echo "<td> <input type='checkbox' name=del[] id=del value=$row[brukerID] size='7'> </td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['brukerID'] . "</td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['Fornavn'] . "</td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['Etternavn'] . "</td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['Telefon'] . "</td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['Epost'] . "</td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['Avgaar'] . "</td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['sms'] . "</td>";
echo "<td onclick=DoNav('medleminfo.php?brukerID=' . $row['brukerID'] . '')>" . $row['dato'] . "</td>";

echo " <td> <a href='rediger.php?brukerID=' . $row['brukerID'] . '' title='rediger'>
<img src='pencil.ico' alt='rediger' height='20px' width='15px' /></a>&nbsp;"; 
?>
<a href="javascript:confirmation (<? echo $row['brukerID'].','.$row['Fornavn']).'&'.$row['Etternavn'].'"'; ?>
" title='slett'><img src='delete.ico' alt='slett' height='15px' width='15px' /></a> &nbsp;
<?
echo "<a href='epost.php?brukerID=' . $row['brukerID'] .
'"title='e-post'><img src='e_mail.ico' alt='epost' height='20px' width='20px' /></a></td></tr>";
}

```

I medleminfo.php vises som sagt all lagret informasjon om medlemmene. Det hentes ut ved å bruke en SQL-spørring og legges inn i en tabell. Er medlemmene medlem av flere enn en gruppe kommer det opp ekstra rader som viser disse gruppene. Se på bildet nedenfor.

MEDLEMINFO	
Fornavn:	Tonje
Etternavn:	Henriksen
Adresse:	Kleoek
Postnr:	319
Sted:	Oslo
Gruppe:	Menneskerettighetsuka
Gruppe:	Regionsgruppen
Epost:	bestizznr1@hotmail.com
Telefon:	456845
Sms:	J
Medlem siden:	2012-04-09
Avgangsår:	2014

Figur: Se medlemsinfo

For å vise alle gruppene et medlem er medlem av utføres det først en SQL-spørring for å sjekke om medlemmet er medlem av noen grupper. Hvis spørringen resulterer i mer enn 0 hentes gruppene ut på samme måte som i nedtrekksmenyen på medlemssiden ved å bruke to SQL-spørringer og to while-løkker. Det eneste som er annerledes er at den her bare skriver ut de gruppene som stemmer overens med brukerID som er satt.

```
$query = mysql_query("SELECT * FROM medlemGrupper WHERE brukerID='$id'");
$result = mysql_num_rows($query);

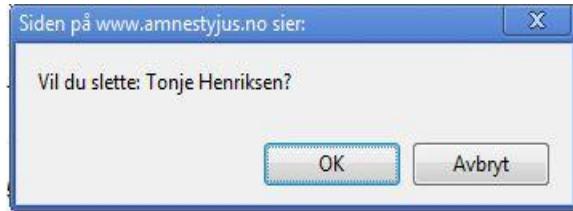
if ($result >0){
    for($i = 0; $i < count($result); $i++){
        //echo $grupper[$i]."<br/>";
        $query = mysql_query("SELECT DISTINCT Tittel from Grupper, medlemGrupper
                            where brukerID='$id' and medlemGrupper.gruppeID=Grupper.gruppeID");
        $query2 = mysql_query("SELECT DISTINCT Tittel FROM medlemGrupper, Undergrupper
                            where brukerID='$id' and gruppeID=gID");

        while ($row = mysql_fetch_assoc($query)) {
?><tr> <td><b> Gruppe: </b></td>
            <td><? echo $row['Tittel'] ?></td></tr><?
        }
        while ($row = mysql_fetch_assoc($query2)) {
?> <tr> <td><b> Gruppe: </b></td>
            <td><? echo $row['Tittel'] ?></td></tr>
            <?
        }
    }
}
?>
```

For å slette et medlem har man to muligheter. Man kan enten trykke på  i valgfeltet på høyre side eller huke av medlemmet for deretter å trykke på slett knappen. På det siste alternativet har man mulighet til å kunne slette flere medlemmer. For å unngå at feil medlemmer blir slettet har vi valgt å bruke JavaScript til å få administrator til å konfirmere at de skal slettes.

For å slette et medlem har vi laget en funksjon som vi kaller på i tabellen. Her har vi med brukerID, fornavn og etternavn. De blir slettet ved å bruke brukerID siden denne er ulik for alle. Fornavn og etternavn er brukt i teksten for å sjekke at det var dette medlemmet som skulle slettes. Trykkes «OK» på konfirmeringsboksen blir de sendt til slettmedlem.php med brukerID hvor koden for å slette medlemmet ligger.

```
function confirmation(brukerID,Fornavn,Etternavn){
var conBox = confirm("Vil du slette: " + Fornavn + " " + Etternavn + "?");
if(conBox){
location.href="slettmedlem.php?brukerID="+ brukerID;
}else{
return;
}
}
```



Figur: Sikkerhetsspørsmål ved sletting

I slettmedlem.php sjekkes det om brukerID er gyldig, er et tall og om det er større enn 0. Videre blir brukeren slettet i alle(tre) tabellene i databasen hvor den vil være lagret. Vi har limit 1 for å unngå at det blir slettet flere medlemmer ved en feiltagelse her. Medlemmet blir altså slettet og sendt tilbake til medlemmer.php.

```
<?php
include "db.php";

$id=$_GET['brukerID'];

if (isset($_GET['brukerID']) && is_numeric($_GET['brukerID']) &&
$_GET['brukerID'] > 0)
{
mysql_query("DELETE FROM medlemGrupper WHERE brukerID='$id'");
mysql_query("DELETE FROM bruker WHERE epost in (select Epost from medlem where
brukerID=$id) LIMIT 1");
mysql_query("DELETE FROM medlem WHERE brukerID='$id' LIMIT 1");

header("Location: medlemmer.php");
}

?>
```

Skal flere medlemmer slettes er det bedre å huke av medlemmene for deretter å trykke på «Slett»-knappen. Også her bruker vi javascriptfunksjonen onclick til å sjekke om disse medlemmene skal slettes, men her har vi valgt å ha koden for konfirmasjonsboksen på «Slett» -knappen. Velges «OK» her blir de sendt til nySlett.php.

```
<input type=submit name='justdel' value='Slett'
onclick="return confirm('Sikker på at du vil slette disse medlemmene?');">
```

I nySlett.php settes det først en variabel med id til de medlemmene som er huket av og en variabel som teller hvor mange som er huket av. Deretter sjekkes det om «Slett» knappen er trykket og hvis den er det går det en for løkke som sletter alle medlemmene. Hvis de blir slettet blir de sendt tilbake til medlemmer.php og hvis det oppstår en feil vises feilen.

```
///slett medlemmer
$id = $_POST[del];
$count = count($id);

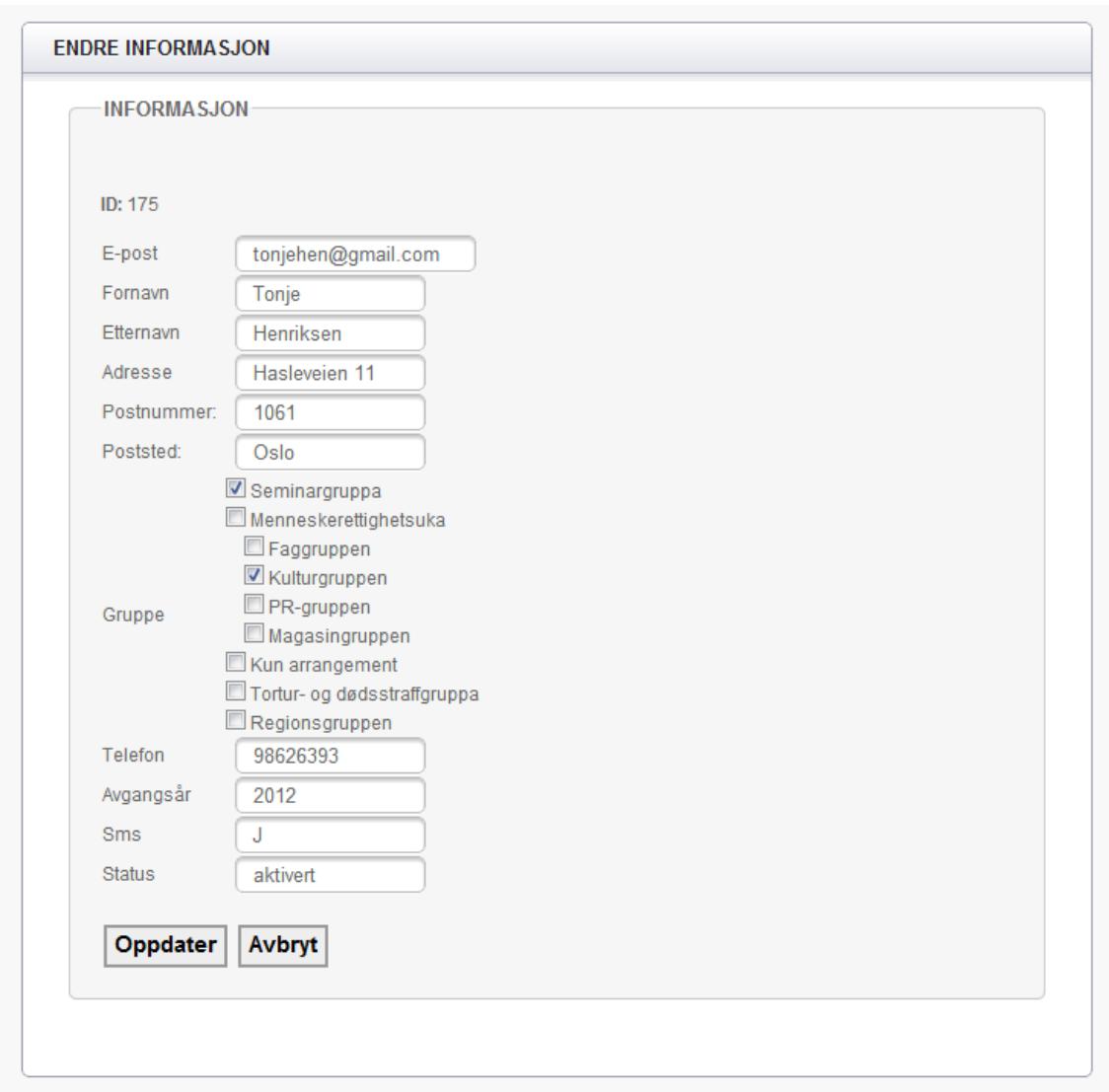
if (isset($_POST['justdel'])){

for ($i=0; $i<$count; $i++)
{
$sql_delete = mysql_query("delete from medlemGrupper where brukerID=$id[$i]");
$sql_delete2 = mysql_query("delete from medlem where brukerID=$id[$i]");
$sql_delete3 = mysql_query("delete from bruker where epost in (select Epost from medlem where brukerID=$id[$i])");
}

if ($sql_delete && $sql_delete2)
{
    header('Location: medlemmer.php');
}
else {
    echo mysql_error ();
}
}
```

Rediger

Trykker man på  i valgfeltet kommer man til rediger.php. Her kan administrator velge å redigere informasjonen om medlemmene. Det kommer opp et skjema likt det som brukes til å legge til et nytt medlem, bortsett fra at all informasjonen allerede er fylt ut. Dermed kan endringene utføres uten at det behøves å fylle ut all informasjon på nytt.



INFORMASJON	
ID:	175
E-post	tonjehen@gmail.com
Fornavn	Tonje
Etternavn	Henriksen
Adresse	Hasleveien 11
Postnummer:	1061
Poststed:	Oslo
Gruppe	<input checked="" type="checkbox"/> Seminargruppa <input type="checkbox"/> Menneskerettighetsuka <input type="checkbox"/> Faggruppen <input checked="" type="checkbox"/> Kulturguppen <input type="checkbox"/> PR-gruppen <input type="checkbox"/> Magasingruppen <input type="checkbox"/> Kun arrangement <input type="checkbox"/> Tortur- og dødsstraffgruppa <input type="checkbox"/> Regionsgruppen
Telefon	98626393
Avgangssår	2012
Sms	J
Status	aktivert

Oppdater **Avbryt**

Figur: Endre brukerinformasjon

For å få opp all informasjonen sjekker vi først om brukerid er gyldig som vi har gjort tidligere. Hvis id er gyldig settes variabelen \$id til brukerID og deretter henter vi ut informasjonen til det medlemmet fra databasen. Hvis det finnes et medlem med den brukerIDen settes det variabler på alle felt som hentes fra databasen og viser deretter skjemaet med all informasjonen allerede fylt inn ved å bruke funksjonen renderform. Hvis det ikke er noen i databasen med den brukerIDen vises det en feilmelding.

```

// få id fra URL, sjekker at det er gyldig
if (isset($_GET['brukerID'])) && is_numeric($_GET['brukerID']) && $_GET['brukerID'] > 0
{
    // query db
    $id = $_GET['brukerID'];
    $result = mysql_query("SELECT medlem.*, Tittel, poststed FROM medlem, medlemGrupper, Grupper,
    post WHERE medlem.brukerID=$id and medlemGrupper.brukerID=$id and medlemGrupper.gruppeID=Grupper.gruppeID");
    or die(mysql_error());
    $row = mysql_fetch_array($result);

    // sjekker at id matcher med en rad i db
    if($row)
    {

        // data fra db
        $id = $row['brukerID'];
        $epost = $row['Epost'];
        $fornavn = $row['Fornavn'];
        $etternavn = $row['Etternavn'];
        $gruppe = $row['Tittel'];
        $adresse = $row['Adresse'];
        $postnr = $row['Postnr'];
        $sted = $row['poststed'];
        $tlf = $row['Telefon'];
        $sms = $row['sms'];
        $avgang = $row['Avgaar'];
        $status = $row['Status'];
        $dato = $row['dato'];

        // vis skjema
        renderForm($id, $epost, $fornavn, $etternavn, $gruppe, $adresse, $postnr, $sted, $tlf, $avgang, $sms, $status, '');
    }
    else
        // ingen match
    {
        echo "Ingen resultat!";
    }
}

```

Hvis knappen oppdater er trykket hentes verdiene fra skjemaet ut og settes i variabler. Vi bruker her PHP funksjonen `mysql_real_escape_string()` som tar bort spesielle tegn siden det settes inn i databasen. Det sjekkes deretter om det er noen felt som er tomme og hvis det er det vises det en feilmelding sammen med skjemaet med de fyltefeltene. Hvis alle feltene var fylt inn oppdateres databasen med den informasjonen som er fylt inn.

```

// sjekker om skjemaet er sendt inn. hvis det har lagre data til db
if (isset($_POST['oppdater']))
{
    // få data fra skjema
    $id = $_POST['brukerID'];
    $epost = mysql_real_escape_string($_POST['epost']);
    $fornavn = mysql_real_escape_string($_POST['fornavn']);
    $etternavn = mysql_real_escape_string($_POST['etternavn']);
    $gruppe = mysql_real_escape_string($_POST['gruppe']);
    $adresse = mysql_real_escape_string($_POST['Adresse']);
    $postnr = mysql_real_escape_string($_POST['postnr']);
    $sted = mysql_real_escape_string($_POST['sted']);
    $tlf = mysql_real_escape_string($_POST['telefon']);
    $avgang = mysql_real_escape_string($_POST['avgang']);
    $sms = mysql_real_escape_string($_POST['sms']);
    $status = mysql_real_escape_string($_POST['status']);

    // sjekk at alle obligatoriske felt er fylt inn
    if ($epost == '' || $fornavn == '' || $etternavn == '' || $adresse == '' ||
    || $postnr == '' || $sted == '' || $tlf == '' || $avgang == '' || $sms == '' || $status == '')
    {
        // feilmeldingen
        $error = '<div id=feilmelding><h3>FEILMELDING:</h3> Vennligst fyll inn de obligatoriske feltene!</div>';

        //error, vis skjema
        renderForm($id, $epost, $fornavn, $etternavn, $gruppe, $adresse, $postnr, $sted, $tlf, $avgang, $sms, $status, $error);
    }
    else
    {
        // lagre data til databasen

        mysql_query("UPDATE medlem, medlemGrupper SET Fornavn='$fornavn',
        Etternavn='$etternavn', Epost='$epost', Adresse='$adresse', Postnr='$postnr',
        Telefon='$tlf', Avgaar='$avgang', sms='$sms', Status='$status' WHERE medlem.brukerID=$id and medlemGrupper.brukerID=medlemGrupper.brukerID");
        or die(mysql_error());
    }
}

```

Trykker man på konvolutten i valgfeltet kommer skjemaet for å sende e-post opp med e-post adressen allerede fylt ut. Alle felt er obligatoriske, og det er også mulig å legge til en fil. Trykkes «Avbryt» kommer man tilbake til medlemmer.php

The screenshot shows a web-based email form titled 'E-POST'. It includes fields for 'Til' (Recipient) containing 'tonjehen@gmail.com', 'Emne' (Subject), a file upload field showing 'Velg fil' (Select file) and 'Ingen fil valgt' (No file selected), and a large 'Melding' (Message) text area. At the bottom are 'Send' and 'Avbryt' (Cancel) buttons.

Siden det er mulig å legge ved en fil i e-posten må vi bruke MIME. MIME indikerer at det er flere deler enn bare en tekst i body i e-posten. Vi setter en variabel med en tilfeldig streng som blir brukt som Boundary marker(grensemarkør). Denne strengen sier fra når teksten i body slutter og den vedlagte filen begynner og slutter.

Videre lagres filinformasjonen i variabler, feltene til og emne lagres i variabler og så lagres selve meldingen som skal sendes. Her har vi lagt meldingen i en tabell og lagt til litt CSS. Fornavnet hentes ut fra databasen og legges til i meldingen, selve teksten skrives ut og tilslutt hvem meldingen er fra. Meldingen lagres i en ny variabel hvor vi bruker funksjonen N12br() som gjør om
 til \n, slik at det blir linjeskift i meldingen.

Det sjekkes så om det er en fil lastet opp og hvis den er det åpnes filen via funksjonen fopen, dataen i filen settes til en variabel, filen lukkes og tilslutt krypteres filen og splittes opp i akseptable setninger.

```

// lager en tilfeldig string til å bli brukt til boundary marker
$mime_boundary="==Multipart_Boundary_x".md5(mt_rand())."x";

// lagrer filinformasjonen til variabler
$tmp_name = $_FILES['vedlegg']['tmp_name'];
$type = $_FILES['vedlegg']['type'];
$name = $_FILES['vedlegg']['name'];
$size = $_FILES['vedlegg']['size'];

$To = $epost;
$Subject = $_POST["emne"];

$message = '<table cellspacing="0" cellpadding="0" width="100%" style="border:1px solid #9CA1B0; background:#FFFFFF;">';
$message .= '<tr><td></td></tr>';
$message .= '<tr> <td width="200" style=" padding:15px 10px 30px 15px; font-size:1em; line-height:1.2em; font-family:arial, georgia,serif;">';
$message .= 'Kjære '. $fname .", <br /><br />" . $tekst;
$message .= '<br /><br /> Med vennlig hilsen <br /> Amnesty Oslo Jus';
$message .= '</td></tr></table>';
$body =nl2br($message);

// sjekker om en fil er lastet opp
if(is_uploaded_file($tmp_name)) {

    // opner filen
    $file = fopen($tmp_name,'rb');
    // leser filen til en variabel
    $data = fread($file,filesize($tmp_name));
    // lukker filen
    fclose($file);
    // krypterer den og splitter den i akseptable linjer
    $data = chunk_split(base64_encode($data));
}

```

Så settes headeren med avsender, mimeversjonen vi bruker, content-type og grensemarkørrene. Så legges meldingen inn i headeren og vedlegget legges med.

Hvis en fil er lastet opp sendes filen sammen med meldingen og emnet, hvis det ikke er lastet opp en fil sendes kun emnet og meldingen. Blir den sendt vises det en vellykket melding på skjermen.

```

$Header = "From: noreply@amnestyjus.no\r\n";
$Header .= "MIME-Version: 1.0\r\n";
$Header .= "Content-Type: multipart/mixed; boundary=\"$mime_boundary\"\r\n\r\n";
$Header .= "This is a multi-part message in MIME format.\r\n";
$Header .= "--{$mime_boundary}\r\n";
$Header .= "Content-type: text/html; charset=iso-8859-1\r\n";
$Header .= "Content-Transfer-Encoding: 7bit\r\n\r\n";
$Header .= $body."<br /><br />";

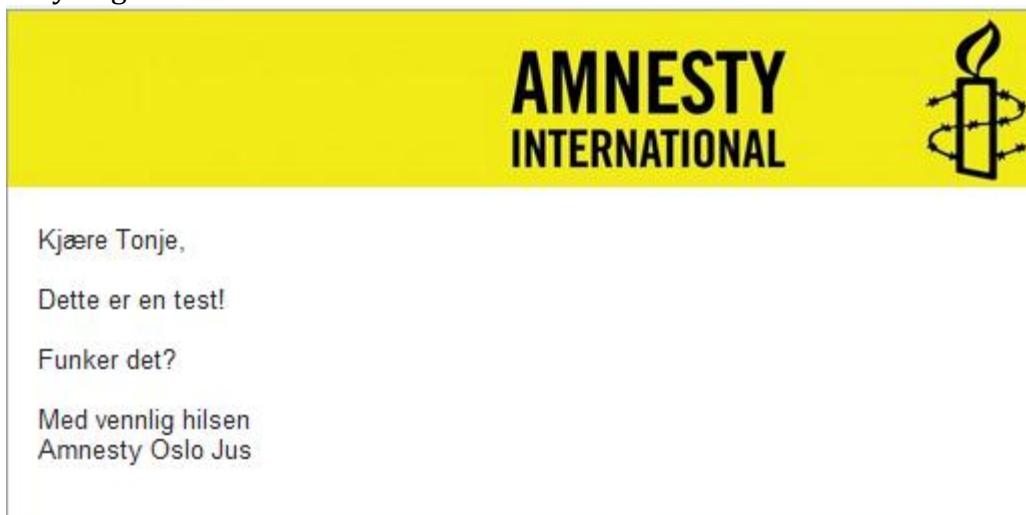
$vedlegg .= "--{$mime_boundary}\n .
"Content-Type: {$type};\n".
" name=\"{$name}\"\n .
"Content-Transfer-Encoding: base64\n\n" .
$data . "\n\n" .
"--{$mime_boundary}--\n";

if(is_uploaded_file($tmp_name)){
    // meldingen sendes
    if (@mail($To, $Subject, $vedlegg, $Header))
        echo "<h4 class='alert_success'>E-post sendt!</h4>";
    else
        echo "Failed to send";
    }
    else
    {
        if (@mail($To, $Subject, null, $Header))
            echo "<h4 class='alert_success'>E-post sendt!</h4>";
        else
            echo "Failed to send";
    }
}

```

 **E-post sendt!**

Dette er et eksempel på en e-post som mottas av medlemmene. Første del av eposten er lik headeren på hjemmesiden til Amnesty og ellers har meldingen en hvit bakgrunn med en tynn grå ramme.



Excel

På toppen av tabellen har vi et excel-ikon . Ved å trykke på denne blir det lastet ned en xls fil med oversikt over alle medlemmene i gruppen. Dette er en enkel fil som kun henter ut medlemmene fra databasen og det er derfor ikke lagt med noe html eller css kode.

Først settes det en funksjon. Hvis et felt allerede inneholder ett eller flere tabulatortegn eller linjeskift vil dette føre til at hele tabellen vil se annerledes ut. For å unngå dette har vi en funksjon som unngår disse tegnene. Før hver rad blir skrevet ut blir tabulatortegnene erstattet med \t og eventuelle linjeskift blir erstattet med \n. Funksjonen sjekker også om det er noen doble fnutter. Funksjonen kalles opp av php funksjonen array_walk() i while-løkken nederst.

Så settes filnavnet: medlemmer og datoen den blir lastet ned pluss filendelsen .xls som sier at det er en excel fil. SQL- spørringen henter ut all informasjon om alle medlemmene som er i den gruppen. Første linje av output vil være kolonneoverskriftene. Verdier skiller seg med en fane \t og rader skiller seg med et linjeskift \n.

```
function cleanData(&$str)
{
    $str = preg_replace("/\t/", "\t", $str);
    $str = preg_replace("/\r?\n/", "\n", $str);
    if(strstr($str, ''')) $str = '''. str_replace(''''', ''''', $str) . '''';
}

# filnavn til nedlasting
$filename = "Medlemmer " . date('d-m-Y') . ".xls";

header("Content-Disposition: attachment; filename=\"$filename\"");
header("Content-Type: text/plain");

$flag = false;
if (isset($_GET['gruppeID']) && is_numeric($_GET['gruppeID']) && $_GET['gruppeID'] > 0)
{
    $id = $_GET['gruppeID'];
    $result = mysql_query("SELECT medlem.brukerID, Fornavn, Etternavn, Adresse, medlem.Postnr, poststed, Telefon, medlem.Epost, Avgaar, sms, Annet, Aktivering, Status, dato, Tittel FROM medlem, medlemGrupper, Grupper, post WHERE medlem.brukerID=medlemGrupper.brukerID AND medlemGrupper.gruppeID='$id' AND medlemGrupper.gruppeID=Grupper.gruppeID AND medlem.Postnr=post.postnr ORDER BY medlem.brukerID") or die(mysql_error());

    while(false != ($row = mysql_fetch_assoc($result))) {
        if(!$flag) {
            // vis felt/kolonne som første rad
            echo implode("\t", array_keys($row)) . "\r\n";
            $flag = true;
        }
        array_walk($row, 'cleanData');
        echo implode("\t", array_values($row)) . "\r\n";
    }
}
```

Legg til medlem

Det er en link til å legge til et nytt medlem på forsiden og på medlemssiden. Skjemaet som kommer opp er da det samme som registreringsskjemaet med den samme valideringen. Det eneste som er annerledes her er at passordfeltet er tatt bort. Istedent lages det et tilfeldig passord som sendes på e-post til det nye medlemmet sammen med informasjonen om at de har blitt medlem.

Denne funksjonen lager et tilfeldig passord på 8 tegn. Først lages en variabel med alle små og store bokstaver i alfabetet sammen med tallene 1-9. Ved å bruke PHP funksjonen

`srand()` blir denne strengen forskjellig hver gang. Variabelen `$i` settes til 0 og variabelen `$pass` som en tom streng. Så lenge `$i` er mindre eller lik 7 så legges et nytt tall til. Til slutt krypteres passordet med `md5()`. Det ukrypterte passordet sendes til det nye medlemmet, mens det krypterte passordet lagres i databasen.

```
//lager et tilfeldig passord
function createRandomPassword() {

    $chars = "abcdefghijklmnopqrstuvwxyzåABCDEFHIJKLMNOPQRSTUVWXYZÆØÅ0123456789";
    srand((double)microtime()*1000000);
    $i = 0;
    $pass = '' ;

    while ($i <= 7) {
        $num = rand() % 33;
        $tmp = substr($chars, $num, 1);
        $pass = $pass . $tmp;
        $i++;
    }

    return $pass;
}

$passord = createRandomPassword();
$passwd = md5($passord);
```

Oppdater

Som tidligere forklart får nye medlemmer som registrerer seg en validerings e-post med en link som må trykkes på for å bli medlem. Alle som registrerer seg blir lagret i databasen selv om de ikke trykker på denne linken og for å fjerne disse medlemmene i databasen har vi laget en oppdateringsknapp nederst på medlemssiden. Denne knappen kan administrator selv velge å trykke på når som helst. Trykkes det på denne knappen blir alle medlemmer som ikke har aktivert medlemskapet sitt fjernet og medlemmer hvor avgangs år har gått ut legges i en ny tabell i databasen kalt «seniormedlemmer».

Først legges en SQL spørring for å finne ut hvor mange som ikke er aktivert i en variabel, hvis det er noen så kjører en for-løkke med samme SQL-spørring med en while-løkke som henter ut brukerID til alle som ikke har aktivert seg som legges i en variabel.

Deretter blir de slettet fra de tre tabellene hvor informasjonen om dem er lagret en etter en til alle har blitt slettet. Tilslutt blir medlemmer med avgangs år satt til i fjor satt inn i tabellen «Seniormedlemmer» og blir deretter slettet i de to andre tabellene. De blir ikke slettet i brukertabellen, slik at de fortsatt har mulighet til å logge inn og endre informasjon eller slette seg selv. Deretter blir man ført tilbake til medlemmer.php.

```

$query = mysql_query("SELECT * FROM medlem WHERE Status not like 'aktivert'");
$result = mysql_num_rows($query);

if ($result){
    for($i = 0; $i < count($result); $i++){
        $query = mysql_query("select brukerID FROM medlem WHERE Status not like 'aktivert'");
        while($row = mysql_fetch_assoc($query)){
            $id=$row['brukerID'];

            $sql_delete = mysql_query("delete from medlemGrupper where brukerID=$id limit 1");
            $sql_delete = mysql_query("delete from bruker where epost in (select Epost from medlem where brukerID=$id) limit 1");
            $sql_delete = mysql_query("DELETE FROM medlem WHERE Status not like 'aktivert'");
        }
    }

    $query = mysql_query("INSERT INTO Seniormedlemmer (brukerID, Fornavn, Etternavn, Adresse, Postnr, Epost,
    Telefon, Avgaar, dato)
    SELECT brukerID, Fornavn, Etternavn, Adresse, Postnr, Epost, Telefon, Avgaar, dato
    FROM medlem
    WHERE Avgaar=".date("Y")."-1");

    $sql_delete = mysql_query("DELETE FROM medlem WHERE brukerID in (select brukerID from Seniormedlemmer)");
    $sql_delete = mysql_query("DELETE FROM medlemGrupper WHERE brukerID in (select brukerID from Seniormedlemmer)");

header("Location: medlemmer.php");

```

3.8.4 Grupper

Grupper.php gir en oversikt over alle gruppene og gir muligheten til å redigere, slette og legge til en ny gruppe. Det er også en egen kolonne til e-post hvor det er mulig å legge til og endre forskjellige epostadresser til de forskjellige gruppene. Gruppene hentes ut fra databasen på samme måte som vi henter ut medlemmene på medlemssiden.

GRUPPER		
Tittel	E-post	Valg
Seminargruppa		
Menneskerettighetsuka		
Kun arrangement		
Tortur- og dødsstraffgruppa		
Regionsgruppen		
Faggruppen		
Kulturgruppen		
PR-gruppen		
Magasingruppen		

Legg til Gruppe

Figur: Redigere grupper

Trykker man på i valgfeltet kommer man til redigergruppe.php hvis gruppen er en hovedgruppe eller redigerundergruppe.php hvis gruppen er en undergruppe. Disse to sidene er helt like bortsett fra at det er forskjellige tabeller i databasen som blir oppdatert. Det kommer opp et skjema hvor det er mulig å endre navnet på gruppen og legge til/endre epost-adressen til denne gruppen. Tittel er allerede fylt ut og dette feltet er obligatorisk. Hvis det allerede hører til en epost-adresse til denne gruppen er også e-post feltet fylt ut, men dette feltet er valgfritt. Trykkes «Avbryt» knappen kommer man tilbake til grupper.php.

ENDRE GRUPPENAVN

ID: 5

Tittel: * Epost:

Oppdater **Avbryt**

Figur: Endre gruppenavn

Skjemaet til å legge til en ny gruppe ser helt likt ut som skjemaet til å endre gruppe. Det går kun an å legge til en ny hovedgruppe. Hvis det skulle gått an å opprette en ny undergruppe måtte vi ha endret på oppsettet på databasen og funnet ut en annen måte å få ut alle gruppene med undergrupper på registreringsskjemaet.

3.8.5 Epost til grupper

Under er skjemaet for å sende e-post til grupper. Det er mulig å sende e-post til en gruppe eller flere grupper samtidig. Hukes det av en gruppe i feltet til høyre legges denne gruppen i «Til»-feltet i skjemaet avgrenset med et komma.

E-POST

Til :

Emne :

Ingen fil valgt

Melding :

GRUPPER

- Seniormedlemmer
- Seminargruppa
- Menneskerettighetsuka
- Faggruppen
- Kulturgruppen
- PR-gruppen
- Magasingruppen
- Kun arrangement
- Tortur- og dødsstraffgruppa
- Regionsgruppen

Figur: Sende epost

Koden her er ganske lik som koden i epost.php. Den eneste forskjellen er at det måtte legges til noe kode for å kunne sende til grupper. For å vise gruppene som hukes av i

«Til»-feltet måtte vi bruke javascript. Vi har onclick på hver checkbox hvor det kalles på funksjonen appendValue() hvis den blir huket av. Denne funksjonen defineres i head hvor verdien fra checkbox hentes ut, det legges til et komma og tilslutt legges det inn i feltet. I skjemaet har vi brukt et tekstfelt (textarea) istedenfor et inputfelt for at dette skal fungere.

```
function appendValue(t, v)
{
    t.value += v +=", ";
}

$query = mysql_query("SELECT gruppeID, Tittel FROM Grupper");
$query2 = mysql_query("SELECT gID, Tittel FROM Undergrupper");
?>
<div class="module little">
    <header><h3>Grupper</h3></header>
    <?php
        $query = mysql_query("SELECT gruppeID, Tittel FROM Grupper");
        while ($row = mysql_fetch_assoc($query)) {
            ?
            <input type="checkbox" name= "gruppe" id="<?=$row[gruppeid]?>" value="<? echo $row[Tittel]?>" onclick="if(checked) appendValue(elements[0], value);" />
            <label for="<?=$row[Tittel]?>"><? echo $row[Tittel]?></label> <br/>
            <?php
                if($row[Tittel] == "Menneskerettighetsuka"){
                    ?
                    <?php
                        $query2 = mysql_query("SELECT gID, Tittel FROM Undergrupper");
                        while ($row2 = mysql_fetch_array($query2)) {
                            ?
                            &nbsp;&nbsp;&nbsp;<input type="checkbox" id="gruppe" value="<?echo $row2[Tittel]?>" name= "gruppe[]" onclick="if(checked) appendValue(elements[0], value);" />
                            <label for="<?=$row2[Tittel]?>"><? echo $row2[Tittel]?></label> <br/>
            
```

Hvis skjemaet sendes lagres de to andre feltene i variabler som vanlig, mens feltet til(Epost) bruker PHP funksjonen explode() som deler opp teksten der det er et komma mellom og legger det i et array. I for-løkka utføres det en SQL-spørring hvor epostadressene til de som er i de forskjellige gruppene hentes ut. Vi bruker her distinct slik at e-posten ikke sendes til samme person flere ganger hvis de er medlem av flere grupper. While-løkken legger alle epostene til en variabel.

```
// Hvis skjema har blitt sendt inn
if (isset($_POST['sendE'])) {
    // data fra skjema
    $emne = mysql_real_escape_string($_POST['emne']);
    $tekst = mysql_real_escape_string($_POST['fritekst']);
    $grupper = explode(", ", $_POST['Epost']);

    for($i = 0; $i < count($grupper); $i++){
        //echo $grupper[$i]."<br/>";
        $query = "SELECT DISTINCT M.Epost, Fornavn FROM Grupper G, medlemGrupper mG,
        medlem M, Undergrupper U WHERE M.brukerID= mG.brukerID AND
        (U.Tittel='$grupper[$i]' AND U.gID = mG.gruppeID)
        OR M.brukerID= mG.brukerID AND (G.Tittel='$grupper[$i]' AND G.gruppeID= mG.gruppeID)";

        $result = mysql_query($query) or die(mysql_error());
        while($row = mysql_fetch_array($result)){
            $epost[] = $row['Epost'];
            $fnavn[] = $row['Fornavn'];
        }
    }
}
```

Her lages headeren i e-posten. Headeren her er lik som i epost.php bortsett fra at feltet BCC legges til. Her bruker vi implode som returnerer elementene i arrayet som gjør at e-

posten blir sendt til alle adressene i arrayet. Adressene i BCC feltet vises ikke i meldingen og det er derfor bare e-postadressen i den første variabelen som vises som mottaker.

```
$To = "noreply@amnestyjus.no";
$Header = "From: noreply@amnestyjus.no\r\n";
$Header .= 'BCC: ' . implode(', ', $epost) . "\r\n";
$Header .= "MIME-Version: 1.0\r\n";
$Header .= "Content-Type: multipart/mixed; boundary=\"$mime_boundary\"\r\n\r\n";
$Header .= "This is a multi-part message in MIME format.\r\n";
$Header .= "--{$mime_boundary}\r\n";
$Header .= "Content-type: text/html; charset=utf-8\r\n";
$Header .= "Content-Transfer-Encoding: 7bit\r\n\r\n";
$Header .= $body."<br /><br />";
```

3.8.6 Statistikk

Statistikk.php gir litt statistikk som kan være litt morsomt å følge med på. Først (til venstre) vises det en oversikt over hvor mange medlemmer som er medlem totalt og hvor mange medlemmer det er i de forskjellige gruppene. Antall medlemmer og hovedgruppene er skrevet i fet skrift, mens undergruppene er skrevet i vanlig tynn skrift. Til venstre er det en oversikt over hvem de fem siste medlemmene som registrerte seg er med fornavn, etternavn og datoen de registrerte seg. Trykker man på raden til et av medlemmene kommer man til siden med oversikt over all lagret informasjon om dette medlemmet (medleminfo.php).

SAMMENDRAG	DE FEM NYESTE MEDLEMMENE		
	Fornavn	Etternavn	Dato
Antall medlemmer: 29			
Regionsgruppen: 7			
Tortur- og dødstraffgruppa: 1			
Seminargruppa: 4			
Menneskerettighetsuken: 8			
Faggruppen: 0			
Kulturgruppen: 1			
PR-gruppen: 1			
Magasingruppen: 1			
Kun arrangement: 1			
	Margit	Haugen	2012-05-16
	Tonje	Henriksen	2012-05-16
	Pernille	Mohn	2012-05-16
	Tonje	Henriksen	2012-05-16
	Margit	Haugen	2012-05-07

Figur: Statistikk

Under er noe av koden for å få frem antall medlemmer. I første SQL spørring spør vi bare om hvor mange medlemmer det er lagret i medlem tabellen og legger dette i en variabel. For å få ut hvor mange som er medlem i de forskjellige gruppene må vi også bruke tabellene medlemGrupper og Grupper. Vi bruker SQL funksjonen count() som returnerer antall verdier. Disse variablene skriver vi senere.

```

$query=mysql_query("Select Count(*) from medlem");
$total = mysql_result($query,0,'count(*');

$query=mysql_query("Select Count(*) from medlem, medlemGrupper,
Grupper where medlem.brukerID=medlemGrupper.brukerID
and medlemGrupper.gruppeID=Grupper.gruppeID and Tittel='Regionsgruppen'");
$totReg = mysql_result($query,0,'count(*');

$query=mysql_query("Select Count(*) from medlem, medlemGrupper,
Grupper where medlem.brukerID=medlemGrupper.brukerID
and medlemGrupper.gruppeID=Grupper.gruppeID and Grupper.gruppeID='4'");
$totToD = mysql_result($query,0,'count(*');

```

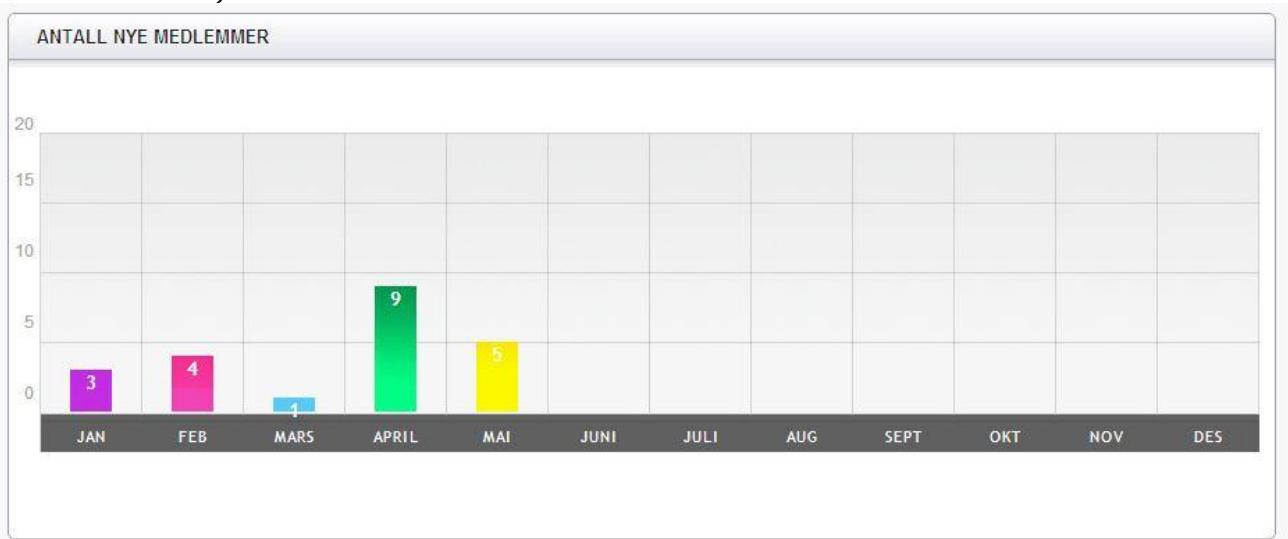
Her vises koden for å få ut de fem nyeste medlemmene. I spørringen sorterer vi medlemmene etter dato med de nyeste datoene først og at bare de fem første vises. Medlemmene skrives ut i en while-løkke på samme måte som i de andre tabellene.

```

<?php
    $result = mysql_query("SELECT * FROM medlem order by dato DESC limit 5")
or die(mysql_error());
?>

```

Her vises det et stolpediagram som viser hvor mange medlemmer som har registrert seg hver måned dette året totalt. Høyden på stolpene varierer etter hvor mange nye medlemmer det er opp til 20 og antallet står oppført øverst i hver stolpe så det blir enkelt å se antallet. Stolpediagrammet viser altså opp til 20 nye medlemmer noe vi synes er passe med tanke på at dette er en skoleorganisasjon og ikke har så mange nye medlemmer hver måned. Derfor er det bedre ha et diagram med færre tall så det er lettere å se forskjell fra måned til måned.



Figur: Statistikk antall nye medlemmer

For å vise antall nye medlemmer i stolpediagrammet må vi først finne ut hvor mange medlemmer som har registrert seg i år og i de forskjellige månedene. Vi bruker SQL funksjonene YEAR(NOW()) som viser det året vi er i nå. Ved å bruke disse funksjonene fungerer diagrammet som det skal hvert år istedenfor bare å fungere et år hvis vi skriver inn et tall. For å få frem antall registrerte hver måned brukte vi funksjonen MONTH() og

tallet på måneden. Denne SQL spørringen er derfor skrevet 12 ganger, 1 for hver måned. Under er tre av spørringene som settes i variabler.

```
//finne antall nye medlemmer i mnd i år  
$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW()) and MONTH(date)='01'");  
$jan = mysql_num_rows($query);  
  
$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW()) and MONTH(date)='02'");  
$feb = mysql_num_rows($query);  
  
$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW()) and MONTH(date)='03'");  
$mar = mysql_num_rows($query);
```

For at stolpene skal få litt høyde ganges hver av de variablene satt ovenfor med 10 og lagres i nye variabler.

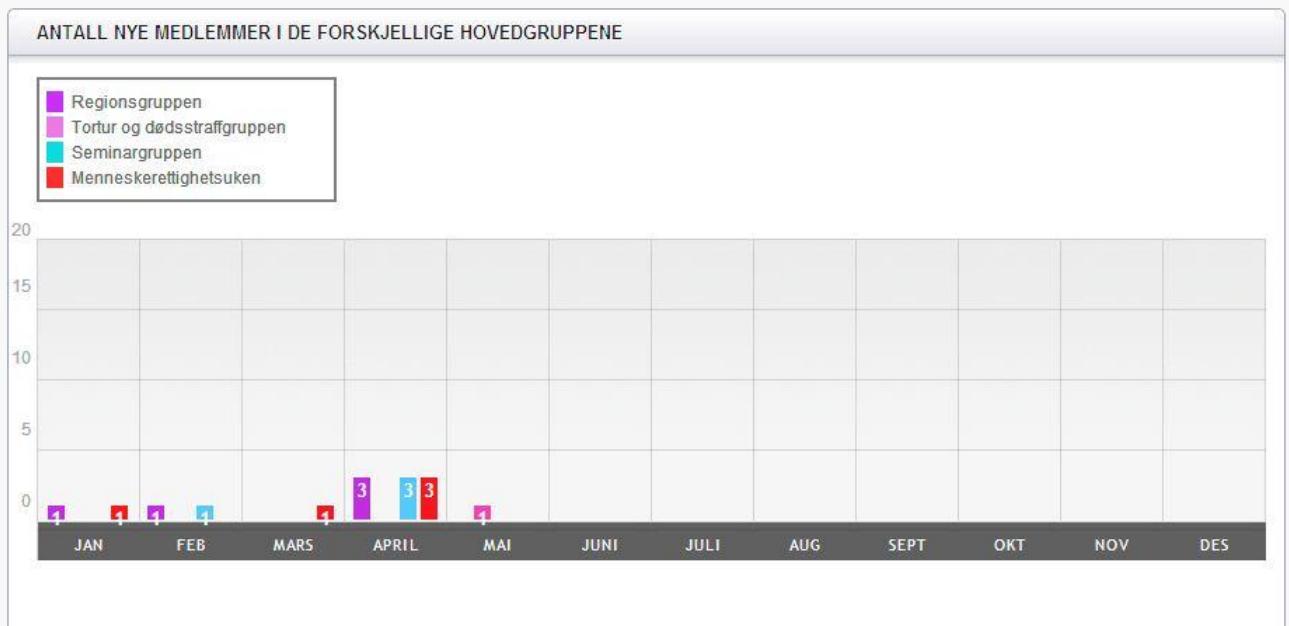
```
//høyden til stolpene i grafen  
$height1=$jan*10;  
$height2=$feb*10;  
$height3=$mar*10;  
$height4=$apr*10;  
$height5=$mai*10;  
$height6=$jun*10;  
$height7=$jul*10;  
$height8=$aug*10;  
$height9=$sep*10;  
$height10=$okt*10;  
$height11=$nov*10;  
$height12=$des*10;
```

Disse nye variablene skrives ut i style og det legges på px (pixel- en verdi for å sette høyde). For å få antallet til å stå skrevet på hver stolpe har vi valgt å bruke en if-setning som skriver ut dette tallet vis det er mer enn 0. Dette var en av tingene vi valgt å ha med etter at vi testet siden på flere nettlesere og fant ut at det ble veldig stygt på nettleseren opera når det ble skrevet ut 0. Vi bruker tre lister for å få oppsettet på diagrammet som vi ønsker. En liste til selve stolpene, en liste til å vise månedene (x-aksen) og en liste til tallene som definerer høyden(y-aksen).

```
//finne antall nye medlemmer i mnd i år  
$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW()) and MONTH(date)='01'");  
$jan = mysql_num_rows($query);  
  
$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW()) and MONTH(date)='02'");  
$feb = mysql_num_rows($query);  
  
$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW()) and MONTH(date)='03'");  
$mar = mysql_num_rows($query);
```

Her vises det hvor mange som har blitt medlemmer hver måned dette året fordelt på de fire hovedgruppene. Øverst vises hvilke farger som er hvilke grupper i

stolpediagrammet.



Figur: Statistikk nye medlemmer i forhold til hovedgrupper

For å finne antall nye medlemmer hver måned dette året fordelt på de fire hovedgruppene gjør vi på samme måte som over bare at vi også kobler inn de forskjellige gruppene. Under viser vi to av de første og andre måned. Det blir her 4x12 sql spørninger hvor vi bare endrer gruppeID og måned.

```
//finne antall nye medlemmer i mnd i år for hver gruppe
//første mnd
$query = mysql_query("SELECT medlem.brukerID FROM medlem, medlemGrupper WHERE YEAR(data)=YEAR(NOW()) and MONTH(data)='01' and medlem.brukerID = medlemGrupper.brukerID and gruppeID=3");
$Gjan3 = mysql_num_rows($query);

$query = mysql_query("SELECT medlem.brukerID FROM medlem, medlemGrupper WHERE YEAR(data)=YEAR(NOW()) and MONTH(data)='01' and medlem.brukerID = medlemGrupper.brukerID and gruppeID=4");
$Gjan4 = mysql_num_rows($query);

//andre mnd
$query = mysql_query("SELECT medlem.brukerID FROM medlem, medlemGrupper WHERE YEAR(data)=YEAR(NOW()) and MONTH(data)='02' and medlem.brukerID = medlemGrupper.brukerID and gruppeID=3");
$Gfeb3 = mysql_num_rows($query);

$query = mysql_query("SELECT medlem.brukerID FROM medlem, medlemGrupper WHERE YEAR(data)=YEAR(NOW()) and MONTH(data)='02' and medlem.brukerID = medlemGrupper.brukerID and gruppeID=4");
$Gfeb4 = mysql_num_rows($query);
```

Høyden på disse stolpene ganges også med 10 for å få litt høyde. Og oppsettet på html er lik her som i diagrammet over bare at det gjøres fire ganger hver måned istedenfor bare en.

```
//høyden til stolpene i grafen
$heightjan1=$Gjan3*10;
$heightjan2=$Gjan4*10;
$heightjan3=$Gjan5*10;
$heightjan4=$Gjan6*10;

$heightfeb1=$Gfeb3*10;
$heightfeb2=$Gfeb4*10;
$heightfeb3=$Gfeb5*10;
$heightfeb4=$Gfeb6*10;
```

```

<div class="stats_graph">

    <div class="spacer"></div><div class="spacer"></div><div class="spacer"></div>
    <div class="spacer"></div><div class="spacer"></div>
<div class="bargraph2" style= "width: 876px;">
    <ul class="bars">
        <li class="bar1a purplebar" style="height:<?php echo $heightjan1; ?>px;"><?if($Gjan3>0){echo $Gjan3;} ?></li>
        <li class="bar1b redbar" style="height:<?php echo $heightjan2; ?>px;"><? if($Gjan4>0){echo $Gjan4;} ?></li>
        <li class="bar1c bluebar" style="height:<?php echo $heightjan3; ?>px;"><?if($Gjan5>0){echo $Gjan5;} ?></li>
        <li class="bar1d crimsonbar" style="height:<?php echo $heightjan4; ?>px;"><?if($Gjan6>0){echo $Gjan6;} ?></li>

        <li class="bar2a purplebar" style="height:<?php echo $heightfeb1; ?>px;"><? if($Gfeb3>0){echo $Gfeb3;} ?></li>
        <li class="bar2b redbar" style="height:<?php echo $heightfeb2; ?>px;"><? if($Gfeb4>0){echo $Gfeb4;} ?></li>
        <li class="bar2c bluebar" style="height:<?php echo $heightfeb3; ?>px;"><? if($Gfeb5>0){echo $Gfeb5;} ?></li>
        <li class="bar2d crimsonbar" style="height:<?php echo $heightfeb4; ?>px;"><? if($Gfeb6>0){echo $Gfeb6;} ?></li>

```

Til slutt sammenliknes antall nye medlemmer i år og i fjor. Den øverste raden viser antall nye medlemmer i fjor og raden nedenfor viser nye medlemmer i år. Det vises to tall i den nederste raden, det første betyr hvor mange flere eller færre nye medlemmer det var i år. Et tallet grønt betyr det at det er flere nye medlemmer i år enn i fjor, et tallet rødt vises det et minustegn foran og betyr at det er færre nye medlemmer i år enn i fjor denne måneden. Tallet i parentes vises hvor mange medlemmer som registrerte seg denne måneden i år.

SAMMENLIKNING ANTALL NYE MEDLEMMER I FJOR OG I ÅR													
ÅR	Januar	Februar	Mars	April	Mai	Juni	Juli	August	September	Oktober	November	Desember	
2011	1	1	2	1	1	1	0	0	0	0	0	0	
2012	2 (3)	3 (4)	-1(1)	8(9)	4(5)	-1(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	

Figur: Statistikk antall nye medlemmer

Først må vi finne antall nye medlemmer hver måned i fjor. Dette gjør vi på samme måte som for å finne i år bare at vi tar funksjonen YEAR(NOW()) og legger til -1 som gir fjoråret.

```

//finne antall nye medlemmer i mnd i fjor
$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW())-1 and MONTH(date)='01'");
$lastJan = mysql_num_rows($query);

$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW())-1 and MONTH(date)='02'");
$lastFeb = mysql_num_rows($query);

$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW())-1 and MONTH(date)='03'");
$lastMar = mysql_num_rows($query);

$query = mysql_query("SELECT * FROM medlem WHERE YEAR(date)=YEAR(NOW())-1 and MONTH(date)='04'";
$lastApr = mysql_num_rows($query);

```

For å finne forskjellen på nye medlemmer i fjor og i år tar vi årets medlemmer minus fjorårets og lagrer dette i nye variabler.

```

//regner ut forskjellen
$diffJan=($jan-$lastJan);
$diffFeb=($feb-$lastFeb);
$diffMar=($mar-$lastMar);
$diffApr=($apr-$lastApr);
$diffMai=($mai-$lastMai);
$diffJun=($jun-$lastJun);
$diffJul=($jul-$lastJul);
$diffAug=($aug-$lastAug);
$diffSep=($sep-$lastSep);
$diffOkt=($okt-$lastOkt);
$diffNov=($nov-$lastNov);
$diffDes=($des-$lastDes);

```

Så lager vi tabellen. I første rad har vi månedene, første kolonne i andre rad skrives det ut årstallet i fjor og i de andre kolonnene skriver vi ut fjorårets resultat. I siste rad skrives først dette årstallet, så skrives forskjellen mellom fjoråret og dette året. Er resultatet over eller lik 0 skrives tallet ut med klassen positiv som gjør at tallet blir grønt, ellers skrives tallet ut med klassen negativ som gjør at tallet blir rødt. Til slutt skrives resultatet fra i år ut i parentes.

```

<div class="centrer">
<table class="tablesorтер" cellspacing="0">
    <thead>
<tr>
<th>ÅR</th>
<th>Januar</th>
<th>Februar</th>
<th>Mars</th>
<th>April</th>
<th>Mai</th>
<th>Juni</th>
<th>Juli</th>
<th>August</th>
<th>September</th>
<th>Oktober</th>
<th>November</th>
<th>Desember</th>
</tr>
    </thead>

```

```

<tbody>
<tr>
<th><strong><? echo date(Y)-1; ?></strong></th>

<td><span><? echo $lastJan; ?></span></td>
<td><span><? echo $lastFeb; ?></span></td>
<td><span><? echo $lastMar; ?></span></td>
<td><span><? echo $lastApr; ?></span></td>
<td><span><? echo $lastMai; ?></span></td>
<td><span><? echo $lastJun; ?></span></td>
<td><span><? echo $lastJul; ?></span></td>
<td><span><? echo $lastAug; ?></span></td>
<td><span><? echo $lastSep; ?></span></td>
<td><span><? echo $lastOkt; ?></span></td>
<td><span><? echo $lastNov; ?></span></td>
<td><span><? echo $lastDes; ?></span></td>
</tr>
<tr>
<th><strong><? echo date(Y); ?></strong></th>
<td>
<span <? if($diffJan>0){ echo '<span class="positif">';} else { echo '<span class="negatif">';} ?><? echo $diffJan; ?>
</span> <small>(<? echo $jan; ?>)</small>
</td>
<td>
<span <? if($diffFeb>0){ echo '<span class="positif">';} else { echo '<span class="negatif">';} ?><? echo $diffFeb; ?>
</span> <small>(<? echo $feb; ?>)</small>
</td>
<td>
<span <? if($diffMar>0){ echo '<span class="positif">';} else { echo '<span class="negatif">';} ?><? echo $diffMar; ?>
</span><small>(<? echo $mar; ?>)</small>
</td>
<td>
<span <? if($diffApr>0){ echo '<span class="positif">';} else { echo '<span class="negatif">';} ?><? echo $diffApr; ?>
</span><small>(<? echo $apr; ?>)</small>
</td>

```

3.8.7 Endre passord

Det går an å endre passordet for å logge inn på administrasjonssidene. For å kunne endre passordet må man kunne det gamle passordet og skrive dette inn først. Deretter skriver man inn det nye passordet to ganger så det ikke blir noen skriveauil hvis det ikke blir noen feil endres passordet når det trykkes på oppdater.

The image shows a user interface for changing a password. At the top, it says "ENDRE PASSORD". Below that are three input fields: "Gammelt passord", "Nytt passord", and "Gjenta passord". At the bottom are two buttons: "Oppdater" (highlighted in grey) and "Avbryt".

Figur: Bilde endre passord.

Hvis knappen «Oppdater» trykkes blir feltverdiene i variabler og det gamle passordet blir kryptert med md5. Så hentes passordet fra databasen ut med en SQL spørring og legges i en variabel. Så hvis passordfeltene er like, det første passordfeltet ikke er tomt og passordet fra databasen er lik med det gamle passordet som er tastet inn krypteres det nye passordet med md5 og databasen oppdateres med det nye passordet. Det kommer tilslutt en melding om at nytt passord er lagret.

```

if (isset($_POST['oppdater']))
{
$passnow = mysql_real_escape_string($_POST['passnow']);
$passnow = md5($passnow);
$pass = mysql_real_escape_string($_POST['pass']);
$pass2 = mysql_real_escape_string($_POST['pass2']);

$result = mysql_query("select pass from admin where brukernavn='admin'");
if($result)
{
$row = mysql_fetch_assoc($result);

$dbpass=$row['pass'];

if (($pass==$pass2) && ($pass!=='') && ($dbpass==$passnow))
{
    $passord = md5($pass);
    // lagre data til databasen
    mysql_query("UPDATE admin SET pass='$passord' WHERE brukernavn='admin'")
    or die(mysql_error());

    echo "Nytt passord lagret";
    exit();
}
}
}

```

Hvis de to passordfeltene ikke er like kommer dette opp som feilmelding, hvis noen av feltene er tomme kommer dette opp som feilmelding og hvis passordet fra databasen ikke stemmer med det som er skrevet inn kommer dette opp som feilmelding. Vi har da altså valgt å ha en egen feilmelding som vises for hver feil slik at administrator lett skal kunne se hva som var feil istedenfor at vi bare skal ta alt i en feilmelding og f.eks. skrive «en feil har oppstått prøv igjen».

```

    if ($pass!=$pass2)
    {
    echo '<h4 class="alert_error">Passordene matcher ikke</h4>';
    }
    if ($passnow == '' || $pass == '' || $pass2 == '')
    {
    echo '<h4 class="alert_error">Minst et av feltene er tomme</h4>';
    }
    if ($dbpass != $passnow)
    {
    echo '<h4 class="alert_error">Feil passord! </h4>';
    }

```

3.8.8 Logg ut



Figur: logg ut .

Linken for å logge ut er plassert på høyre side under Amnesty logoen. Koden for å logge ut er ikke så komplisert. Først startes session, så ødelegges session, så settes cookie med brukernavnet til minus den samme tiden som ble satt i login.php. Så tilslutt blir administrator ført tilbake til login.php.

```
<?php
    session_start();

    session_destroy();

    setcookie("brukernavn","",time()-60*60*24*100);

    header("Location: login.php");
?>
```

3.9 Testing av systemet

Administratorsiden og medlemssiden ble dessverre litt forsinket og det ble derfor vanskelig å få til et møte med oppdragsgiver for testing da de er studenter midt i eksamenstiden på dette tidspunktet. Vi spurte om de hadde anledning til å møte oss, men da dette ikke gikk sendte vi linkene til administrator, registreringsskjemaet og medlemssiden på e-post med instruksjoner og spurte om de kan teste siden selv og komme med tilbakemeldinger. Vi har selv testet alle sidene to ganger hver for oss for å se om alt fungerer som det skal.

På den første testen kom det frem noen små feil som vi har måttet rette opp.

Administratorsiden

- Informasjonen om noen medlemmer kom ikke frem på medleinfo.php. Dette hadde vi ikke hatt problemer med før og forstod ikke helt hva dette kunne skyldes. Det viste seg tilslutt at et av gruppemedlemmene hadde gjort en endring i innstillingen til databasen som førte til denne feilen og koden ble derfor endret slik det nå fungerer.
- På statistikken var det en av gruppene som ikke viste riktig antall medlemmer.
- Det kom frem at vi burde ha med en side hvor det går an å endre passordet.
- Seniormedlemmer hadde blitt litt glemt. Seniormedlemmene ble ikke vist på noen av sidene og vi bestemte etter hvert at vi ville få frem medlemmene av denne gruppen på samme måte som i de andre gruppene.
- Vi testet logg inn for SQL-injections: Vi skrev inn «1' or '1' = '1» i både brukernavn- og passordfeltet. Vi ble ikke logget inn så denne fungerer som det skal.
- Når vi testet siden på internett Explorer ble utseende litt annerledes, det kom blant annet opp en svart strek rundt alle linker(inkludert bilder).

Registreringsskjemaet

- Feltvalideringen var ikke helt slik vi ville. Feilmeldingen på passordfeltet ble endret samtidig som vi endret telefonvalideringen til å måtte bestå av 8 tall.

Medlemssiden

- Det gikk an å logge seg inn selv om brukeren ikke hadde validert seg.
- Brukeren ble slettet ved å trykke på «meld meg ut»- linken uten noen form for bekreftelse.
- Alle gruppene som medlemmet var medlem av ble ikke vist, det var kun den siste gruppen som ble vist.
- Det kom ikke opp en vellykket melding når medlemmet hadde oppdatert informasjonen om seg selv.

Alle feilene ble rettet opp og sidene ble testet på nytt. Denne gangen var det ingen store feil som ble oppdaget av oss.

3.10 Endringer

Vi har gjort en del endringer på administratorsiden etter hvert som vi har sett at den løsningen vi først gjorde ikke fungerte optimalt. Administratorsiden hadde først lik layout som hovedsiden. Amnesty endret plutselig hele layouten på siden sin og dette førte til noen problemer. De valgte en layout som ikke fulgte alle standarder i Wordpress, det ble flere div-tagger, noen div-tagger endret navn og hele siden ble mye smalere. Det førte til en del problemer både på administratorsiden, registreringsskjemaet og medlemssiden. Layouten på administratorsiden ble tilslutt endret til å bli helt uavhengig av den layouten Amnesty nå har på hovedsiden, mens de andre sidene etter litt om og men fikk til riktig oppsett.

The screenshot shows the homepage of the Amnesty International Juridisk Studentnettverk website. The header is red with the organization's name in white. Below the header is a black navigation bar with links: Home, Om ADS, Gruppene, Kontakt oss / Bli medlem, Vedtekter/referat, and Aktivitetskalender. To the right of the navigation is a stylized candle icon. The main content area has a white background. On the left, there is a section titled "Kontakt oss / Bli medlem" with social sharing buttons (Facebook Like, Send, Tweet, etc.) and contact information. On the right, there are two boxes: "FERSKE INNLEGG" containing recent posts and "KOMMENDE HENDELSER" showing upcoming events with a calendar view.

Den gamle layouten til Amnesty

The screenshot shows the homepage of the Amnesty International Jus Oslo website. The header is yellow with the organization's name in black. Below the header is a black navigation bar with links: Home, Aktivitetskalender, Gruppene, Kontakt oss / Bli medlem, and Om Amnesty Jus Oslo. To the right of the navigation is a stylized candle icon. The main content area has a white background. On the left, there is a section titled "Kontakt oss / Bli medlem" with contact information. On the right, there are two boxes: "Ferske innlegg" containing recent posts and "Kommende hendelser" showing upcoming events with a calendar view.

Den nye layouten til Amnesty

Vi har også gjort noen andre små endringer på sidene gjennom prosjektet.

Gruppevisningen ble endret – til å begynne med skrev vi alle gruppenavnene rett inn i koden og hadde dette som linker, men dette endret vi til å hente gruppene ut fra databasen og legge de i en nedtrekksmeny sånn at det vil fungere selv om gruppenavnene blir endret. Som følge av dette ble det også kluss med sidetallene og dette måtte også endres, noe som viste seg å ta en god stund.

Vi hadde først all informasjon om medlemmene i en tabell, men denne ble etter hvert for lang og vi valgte derfor å ha en egen side med all informasjon og bare ha den viktigste informasjonen i tabellen.

4. Mobilapplikasjon

4.1 Android

Android er en svært omfattende åpen kildekode plattform for mobile enheter. Android inneholder store mengder rammeverk og verktøy som er åpent for alle som ønsker å utvikle applikasjoner for Android.

Applikasjonens rammeverk er et miljø som inneholder flere Java bibliotek, spesielt bygget for Android, det inneholder også flere systemer som gjør tjenester som er i telefonen tilgjengelig for applikasjonen, slik som lokasjon, trådløst og mobile nettverk, sensorer, telefoni samt mange andre.

4.1.2 Java

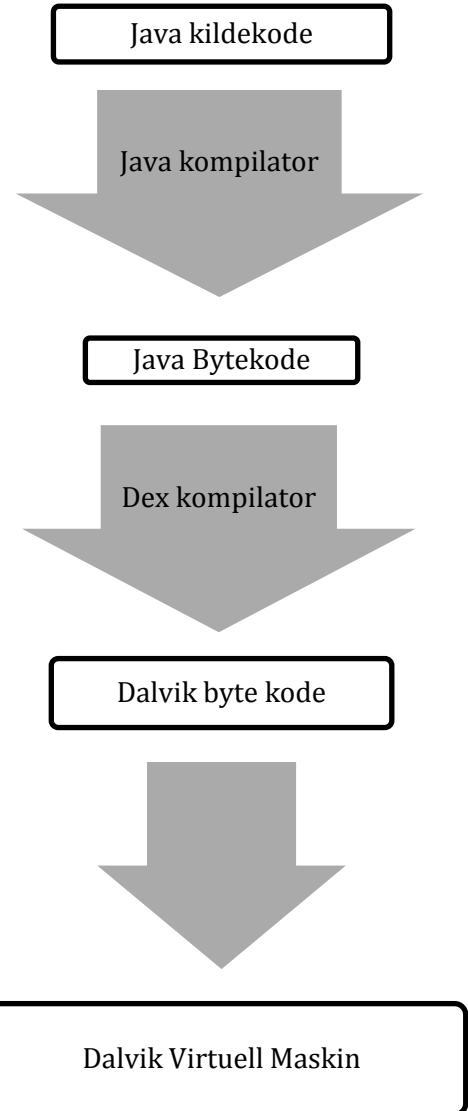
Alle Android programmer blir skrevet i programmeringsspråket Java, som er et objekt orientert språk. Ved programmering i Java, blir Javakoden kompilert til en plattformuavhengig bytekode som kan kjøres i Javas virtuelle maskin. Det som er annerledes ved kompileringen til Android er at Javakoden blir først kompilert til Java bytekode, men deretter blir den kompilert en gang til via Dalvik kompilatoren, slik at det blir omgjort til Dalvik bytekode.

Når det er omgjort til Dalvik bytekode kan dette vises i en virtuell maskin for Android.

4.1.3 eXtensible Markup Language (XML)

XML er et tekstbasert format for visning av strukturert informasjon. Oppsettet for XML er lignende HTML, men reglene er strengere. Dersom koden ikke kompilerer vil heller ikke teksten vises, en feilmelding vil vises i stedet. I Android blir XML brukt til å sette strenger, utseende for de forskjellige skjerm bildene og for hoved oppsettet til hele applikasjonen.

Hver javaklasse vil ha en egen XML fil, som viser til layouten for skjerm bildet som skal vises. Denne hentes via den offentlige metoden `setContentView(R.id.xmlfil)`.



Figur: Kompilering til virtuell Maskin

4.1.4 SQLite

Android kommer med en innebygget database kalt SQLite, dette er en åpen kildekode database. Det er et sett av bibliotek som tilbyr database funksjonene, som blir importert med android.database.sqlite. Databasen blir lagret i en fil, det er altså ingen server som databasen kjøres på.

Android sitt rammeverk har en innebygget klasse kalt SQLiteOpenHelper, denne klassen hjelper til med å lage den lokale databasen.

4.2 Applikasjonens oppbygning

En applikasjon er avhengig av visse filer og mapper for å kunne kjøre, når et nytt prosjekt for en applikasjon blir opprettet vil en del filer og mapper som applikasjonen er avhengig av bli opprettet.

4.2.1 Manifestfilen

Den første grunnleggende filen er manifestfilen, det er denne filen som holder hele applikasjonen sammen. Denne filen må kalles AndroidManifest.xml. I manifestet ligger informasjon om applikasjonen, som systemet må ha før det kan kjøre noe av applikasjonens kode.

Manifestet viser informasjon om

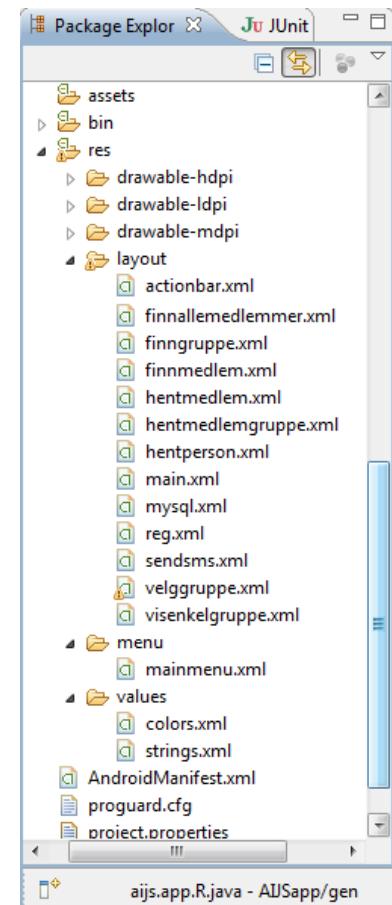
- Android versjon applikasjonen er bygget mot.
- Viser hvilke tillatelser applikasjonen har til andre deler av enheten den kjøres på, slik som tilgang til internett og mulighet for å sende og motta SMS.
- Navn og logo for applikasjonen
- Aktiviteter som skal vises i applikasjonen

4.2.2 Layout

Mappen Layout inneholder XML filer som spesifiserer layouten til skjermbildene som applikasjonen skal vise. Her settes strukturen for skjermbildet og inneholder alle elementer som skal vises på skjermen. Fordelen med denne måten å lage koden for layout er at koden for applikasjonen vil bli fordelt i flere filer. Koden for utseende vil være i én fil, mens koden for hvordan applikasjonen skal oppføre seg blir liggende i en annen fil.

4.2.2.1 Unntak

Dersom utseende på applikasjonen skal endres i forhold til javakode, kan layout legges til dynamisk direkte i kodden. Da kan den offentlige metoden addview() brukes. Dette brukes for eksempel når det skal legges til knapper eller lister i forhold til hvor mange linjer det er i en database. Når layout blir laget direkte i javakoden, vil det ikke være nødvendig med en ekstern XML fil for layout.



Figur: Oversikt over mappen

4.2.3 Values

Mappen Values inneholder automatisk filen strings.xml, denne filen inneholder all tekst som skal vises i applikasjonen. Her kan navn på knapper, overskrifter, tekst som skal vises og lignende. Denne filen hjelper også til med å dele applikasjonen, slik at teksten og navn er lagret her, mens hvordan dette skal fremvises ligger i layoutfilen.

Mappen values kan også inneholde flere filer som kan fremvise hva en forskjellig tekst eller et ord skal representere. Disse filene kan opprettes etter behov, for eksempel opprettelsen av en fil kalt farger.xml i denne mappen. I filen farger.xml kan det legges inn HEX eller RGB koder, som da kan få et navn som kan brukes senere:

```
<color name="skriftfarge">#000000</color>
```

Her er HEX-koden til fargen hvit gitt navnet «skriftfarge», i andre XML filer i applikasjonen kan det derfor henvises til «skriftfarge» for å få frem fargen hvit (#000000).

4.2.4 R.java

R klassen er bindeleddet mellom Java og ressursfilene som ligger i mappen res. Ved sletting, endring eller oppretting av nye filer i denne mappen vil det bli gjort endringer i R.java automatisk. Denne filen skal ikke modifiseres manuelt. R.java kan se ut som dette:

```
public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int action_about=0x7f020000;
    }
    public static final class layout {
        public static final int actionBar=0x7f030000;
        public static final int admin=0x7f030001;
    }
    public static final class string {
        public static final int SMS=0x7f05000f;
        public static final int SMStil=0x7f050010;
    }
}
```

4.2.5 Java kildekode

Javakoden som blir opprettet er den delen som bestemmer hvordan applikasjonen skal oppføre seg.

4.2.6 JavaScript Object Notation (JSON)

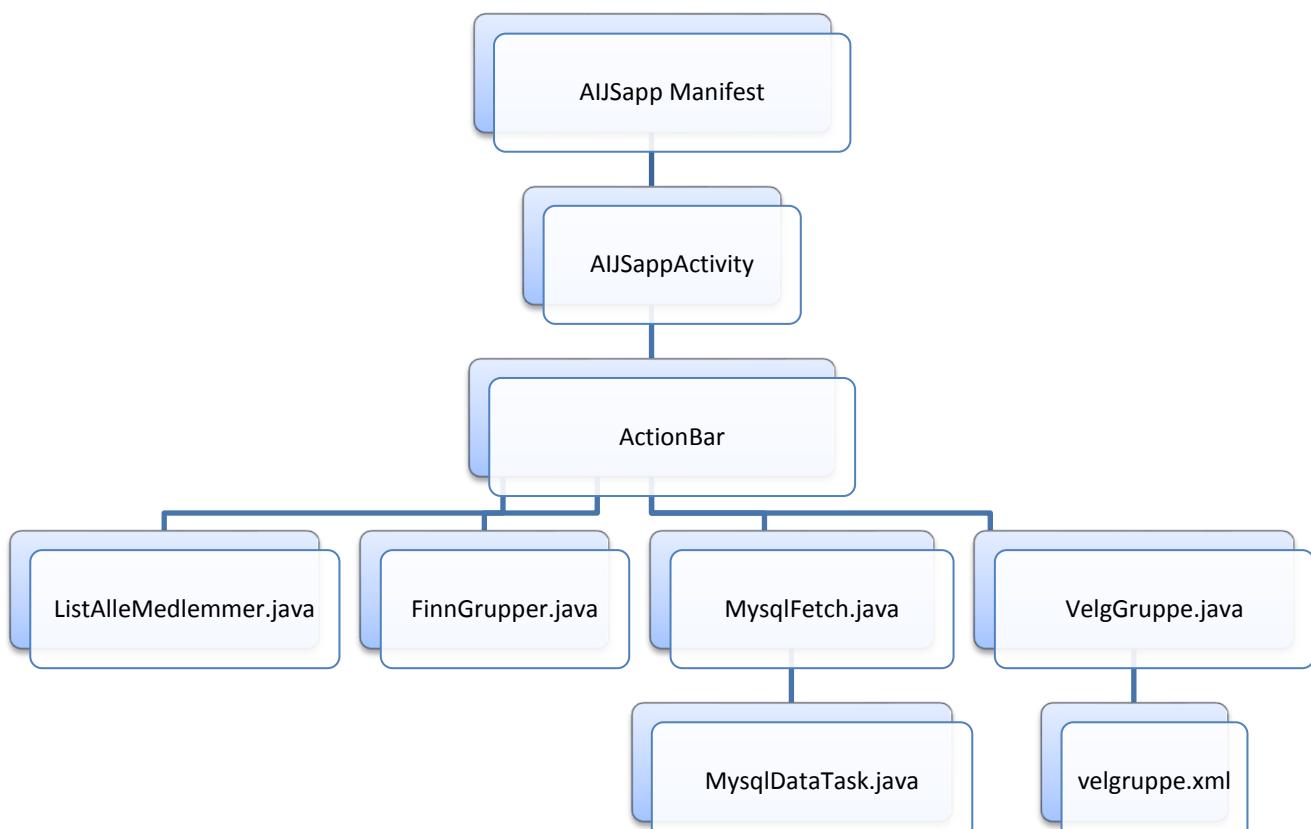
Tekstbaser standard for datautveksling som er basert på JavaScript. Det blir brukt for å sende informasjon mellom en server og applikasjon.

JSON er bygget opp av to strukturer, den første er å sette opp par mellom navn og verdi, der det i forskjellige programmeringsspråk blir brukt som objekt, hashtabell, assosiative matriser og flere andre. Den andre strukturen er ordnet liste av verdier, de fleste språk realiserer dette som en matrise, vektor, liste eller en sekvens.

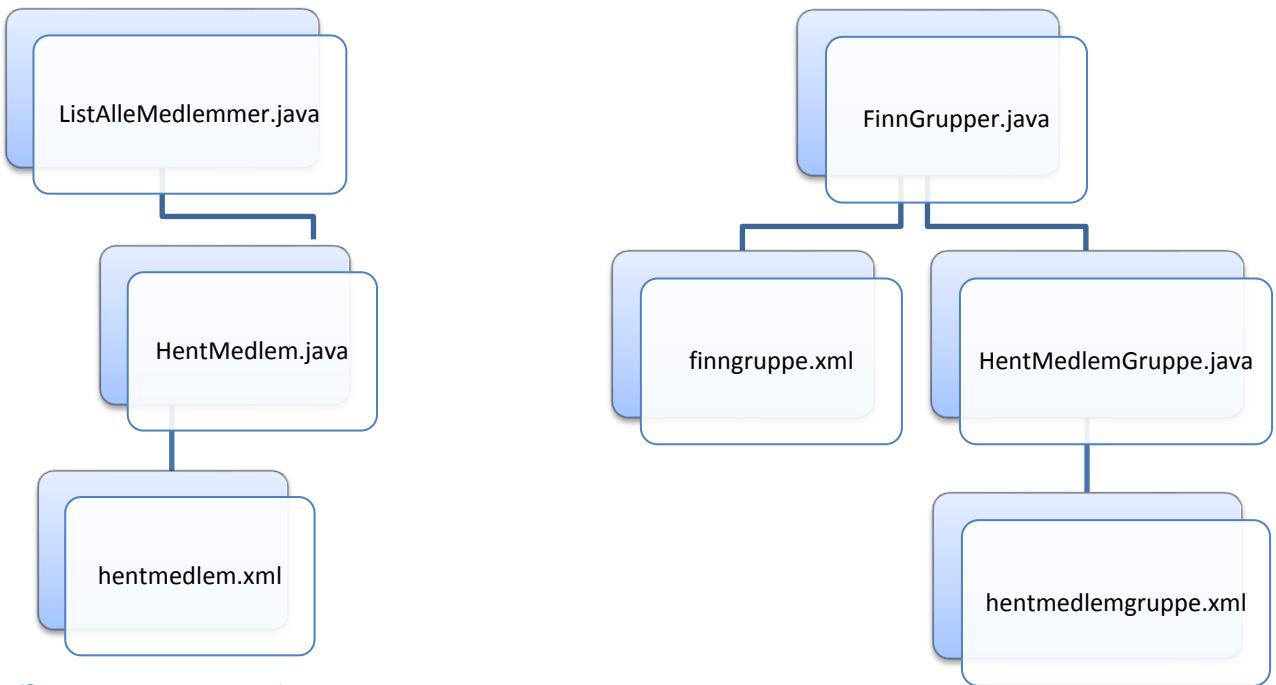
I våres applikasjon blir JSON brukt mellom MySQL server og Android applikasjonen sin lokale database.

4.3 Applikasjonen AIJSapp

4.3.1 Applikasjonskart



Figur - Kart hoveddel av applikasjon



Figur: Oversikt over mappen res 1

Oversikt over mappen res 2

4.3.2 Manifestet

Versjonen for applikasjonen vises i manifestet, sammen med Android versjonen som blir brukt. Den ønskede versjonen er API nivå 15, men minimums versjon er API nivå 9. Versjon 15 går til plattformnivået 4.0.3 - 4.0.4, som også går under navnet «Ice Cream Sandwich». Her kommer det også frem hvilken versjon av XML som skal brukes, og hvilken encoding som skal brukes.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="aijs.app"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="9"
        android:targetSdkVersion="15" />

```

I manifestet kommer det også frem hvilke tillatelser som trengs for å bruke alle deler av applikasjonen, vår applikasjon trenger internett for å koble opp i mot MySQL-databasen, i tillegg til å sende SMS. Derfor er disse tillatelsene også lagt til i Manifestet

```

<uses-permission android:name="android.permission.INTERNET" >
</uses-permission>
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />

```

I den resterende delen av manifestet, <application> elementet, inneholder mer informasjon om selve applikasjonen. Den første informasjonen som blir satt her er

logoen som blir hentet fra mappen drawable, og navnet på applikasjonen som blir hentet via en streng, som er lagret i strings.xml.

Den første aktiviteten som blir oppgitt under application er hovedaktiviteten, den første klassen som skal vises ved oppstart. Vår hovedaktivitet, AIJSappActivity, blir satt som hovedaktivitet via intentfilteret, som setter aktiviteten til *android.intent.action.MAIN*, og *android.intent.category.LAUNCHER*.

Til hver aktivitet er også android:uiOptions="splitActionBarWhenNarrow", dette gjør at menyen vil komme nederst på hver side.

Videre følger de resterende aktivitetene som skal vises i applikasjonen, underliggende klasser som ikke har skjermbilder vises ikke i manifestet.

```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity
        android:name=".AIJSappActivity"
        android:label="@string/app_name"
        android:uiOptions="splitActionBarWhenNarrow" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".ListAlleMedlemmer"
        android:uiOptions="splitActionBarWhenNarrow" >
    </activity>
    <activity
        android:name=".FinnGrupper"
        android:uiOptions="splitActionBarWhenNarrow" >
    </activity>
    <activity
        android:name=".MysqlFetch"
        android:uiOptions="splitActionBarWhenNarrow" >
    </activity>
    <activity
        android:name=".HentMedLemGruppe"
        android:uiOptions="splitActionBarWhenNarrow" >
    </activity>
    <activity
        android:name=".SendSMS"
        android:uiOptions="splitActionBarWhenNarrow" >
    </activity>
    <activity
        android:name=".HentMedLem"
        android:uiOptions="splitActionBarWhenNarrow" >
    </activity>
</application>

</manifest>
```

4.3.3 Hovedaktiviteten

Den første klassen som blir hentet er den minste klassen i applikasjonen. Det som vises i skjermbildet til denne aktiviteten er en velkomsttekst til brukeren, samt en menylinje nederst på skjermen.

Bakgrunnsfargen, tekststørrelsen og velkomstteksten blir hentet via XML filen main.

Menyen nederst på skjermen blir arvet fra klassen ActionBar, som vi vil ta for oss senere.

```
package aijs.app;

import android.os.Bundle;

public class AIJSappActivity extends ActionBar {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

    }
}
```



Figur: Hovedaktivitet

4.3.4 Menylinjen

Menylinjen blir bestemt i klassen ActionBar, her bestemmes hvilke knapper som skal være med og hvor disse knappene skal sende brukeren når de trykker på en knapp i menyen. Innenfor metoden onOptionsItemSelected vises de forskjellige knappene innenfor kontrolltypen switch case. Her vises hvilket ikon som skal brukes for knappen, og den lokale metoden som skal kjøres med knappen.

```
@Override
public boolean onOptionsItemSelected(MenuItem item){
    switch(item.getItemId()){
        case R.id.refresh:
            refresh();
            return true;
        case R.id.SMSGruppe:
            sms();
            return true;
        case R.id.gruppe:
            hentGrupper();
            return true;
        case R.id.VelgGruppe:
            velgGruppe();
            return true;
        case R.id.Info:
            info();
            return true;
    }
}
```

```

        }
        return true;
    }
}

```

Metodene i klassen er refresh(), sms(), hentGrupper(), velgGruppe() og info(). Infoknappen, som bruker metoden info(), vil ikke sende brukeren til noen ny aktivitet, men vil vise brukeren en melding via Toastvarsling. Toastvarsling er en popup som vises i overflaten på skjermen for brukeren.

```

public void info(){
    Toast.makeText(getApplicationContext(), "Denne
        applikasjonen er kun ment for" + " styret for
        Amnesty Jus Oslo, og inneholder medlemsinformasjon "
        + "som kun skal være tilgengelig for styrets
        medlemmer.", Toast.LENGTH_LONG).show();
}

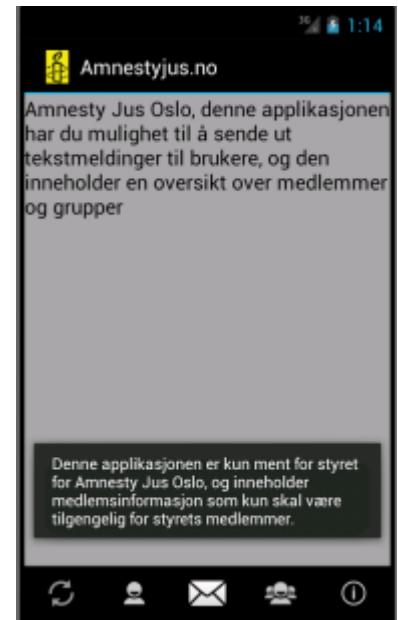
```

De andre metodene sender brukeren videre til en ny aktivitet, dette gjøres via den offentlige metoden startActivity(). Metoden velgGruppe() er et eksempel på dette, når denne metoden blir kalt vil brukeren bli sendt videre til klassen FinnGrupper

```

public void velgGruppe(){
    Intent myIntent = new Intent(getApplicationContext(), FinnGrupper.class);
    startActivity(myIntent);
}

```



Figur: Menylinjen



Oppdatering, denne knappen vil oppdatere den lokale databasen slik at informasjonen den inneholder er lik MySQL databasen.

Medlem, denne knappen vil sende brukeren videre til aktiviteten som viser alle medlemmer som er medlem i Amnesty Jus Oslo

SMS, knappen sender brukeren videre til aktiviteten som gir brukeren mulighet til å sende tekstmeldinger til alle brukerne, eller velge ut en spesifikk gruppe den ønsker å sende tekstmelding til.

Velg gruppe, knappen sender brukeren til aktiviteten der de kan velge ønsket gruppe og se på medlemmene som er medlem i valgt gruppe.

Infoknappen vil vise brukeren kort informasjon om applikasjonens formål.

4.3.5 Oppdatering av databasen

Oppdateringen av databasen er delt i tre deler, og her brukes JSON, MySQL og Java for å få hentet informasjon fra den eksterne databasen til den lokale databasen som bruker SQLite.

Tre PHP filer er lagret på serveren der den eksterne databasen ligger, disse henter ut informasjonen som er nødvendig for applikasjonen via MySQL-setninger. Deretter blir det laget et array som inneholder all informasjonen som er hentet via MySQL. Det siste som blir gjort i disse filene er å gjøre om arrayet til en JSON representasjon av informasjonen.

```
$sql=mysql_query("select brukerID, Fornavn, Etternavn, Telefon, Adresse, Postnr,
                  Epost, Avgaar
                  FROM medlem");
while($row=mysql_fetch_assoc($sql))
$output[]=$row;
print(json_encode($output));
```

Output ifra json_encode vil vises som array og en linje vil se ut som dette:

```
[{"brukerID": "225", "Fornavn": "Tonje", "Etternavn": "Henriksen", "Telefon": "898687976",
,"Adresse": "Hasleveien
11", "Postnr": "0571", "Epost": "hiuj@hotmai.com", "Avgaar": "2014", "sms": "J"}]
```

Det er klassene MysqlFetch og MysqlFetchDataTask som vil hente informasjonen til applikasjonen fra PHP filene.

MysqlFetch vil sende informasjonen om hvilke PHP-sider den ønsker å hente informasjon ifra til MysqlFetchDataTask, som vil hente ut informasjonen og legge det inn i et array og sende dette tilbake til MysqlFetch.

I MysqlFetch vil dataene bli lagt inn i metoder som blir hentet via klassen DBAdapter. I denne filen ligger alle metoder for den lokale databasen og vi vil ta for oss hele denne klassen senere.

MysqlFetch bruker metodene updatePersoner(), updateGrupper() og updateMedlemGrupper(), disse har samme funksjon, men vil plassere de forskjellige arrayene inn i riktig tabell i den lokale databasen.

For å hente de forskjellige gruppenavnene og gruppeIDene, vil metoden updateGrupper() brukes. Her hentes navnene fra arrayet som JSON representasjonen er lagt inn i, og det blir så plassert inn i tabeller som er definert i DBAdapter klassen.

```
db.updateGrupper();
for (int i = 0 ; i < array2.length(); i++) {
    String tittel = array2.getJSONObject(i).getString("Tittel");
    String gruppeid = array2.getJSONObject(i).getString("gruppeID");
    ContentValues cv=new ContentValues(2);
    cv.put(DBAdapter.getTittel(), tittel);
    cv.put(DBAdapter.getGruppeID(), gruppeid);
    db.insert1(cv);
}
```

4.3.6 Lokal database

Det brukes SQLite i den lokale databasen og den inneholder tre forskjellige tabeller, disse er personer, grupper og medlemgrupper. Innholdet i disse tabellene er tilrettelagt for hva som skal vises i applikasjonen. Siden ingen informasjon skal bli oppdatert fra disse tabellene eller informasjon skal endres er de ikke på tredje normalform og kan inneholde dobbeltlagring.

4.3.6.1 Opprettelse av tabeller

Tabellene vil bli opprettet ved første kallet på DBAdapter, onCreate() metoden vil lage tabellene slik som tabellen personer er opprettet her:

```
static final String TABELLPERSONER = "personer";
static final String TABELLGRUPPER = "grupper";
static final String TABELLMEDLEMGRUPPER = "medlemgrupper";
```

```
public void onCreate(SQLiteDatabase db) {
    Log.d(TAG, TABELLPERSONER);
    String sql = "create table " + TABELLPERSONER + " (" + ID
            + " integer primary key autoincrement, "
            + "getBrukerID()"+ " text, "
            + "getFornavn()"+ " text, "
            + "getEtternavn()"+ " text, "
            + "getTelefon()"+ " text, "
            + "getAdresse()"+ " text, "
            + "getPostnr()"+ " text, "
            + "getPost()"+ " text, "
            + "getAvgaar()"+ " text, "
            + "getSMS()"+ " text);";
    Log.d(TAG, "oncreated sql" + sql);
    db.execSQL(sql);
}
```

4.3.6.2 Oppdatering av databasen

Ved oppdatering av databasen vil metodene updatePersoner(), updateGrupper() og updateMedlemGrupper() bli brukt. Hver gang en bruker skal oppdatere tabellene og de allerede har eksisterende tabeller med samme navn, vil denne bli slettet og all informasjon vil bli lagt inn på nytt. Dette er for ikke å få flere like linjer, og dersom et medlem har oppdatert sin informasjon, vil også dette bli oppdatert i applikasjonen.

updateGrupper(), vil slette tabellen grupper, for så å legge til alle gruppeIDer og titlene på gruppene på nytt.

```
public void updateGrupper(){
    db.execSQL("drop table if exists " + TABELLGRUPPER);
    Log.d(TAG, "updated");
    db.execSQL("create table if not exists " + TABELLGRUPPER + " (" + ID
            + " integer primary key autoincrement, " + getTittel()
            + " text, " + getGruppeID() + " text);");
    Log.d(TAG, "oncreated sql");
}
```

4.3.6.3 Hente informasjon fra databasen

For å hente ut informasjon fra databasen brukes grensesnittet Cursor, dette gir et utvalg i en tilfeldig rekkefølge av tabellen som er valgt.

Det er laget flere metoder for å hente informasjon fra databasen, dette er for å finne medlemmer i databasen, finne grupper, og hvem som er medlemmer i de ulike gruppene. Når det blir sendt et kall på metoden finnGrupper() vil denne gi informasjon med tittelen på gruppen, og hvilken gruppe ID denne gruppen har.

```
public Cursor finnGrupper() {  
    Cursor cur;  
    String[] cols = { getTittel(), getGruppeID() };  
    cur = db.query(TABELLGRUPPER, cols, null, null, null, null,  
    getTittel());  
    return cur;  
}
```

4.3.7 Meldingsutsendig

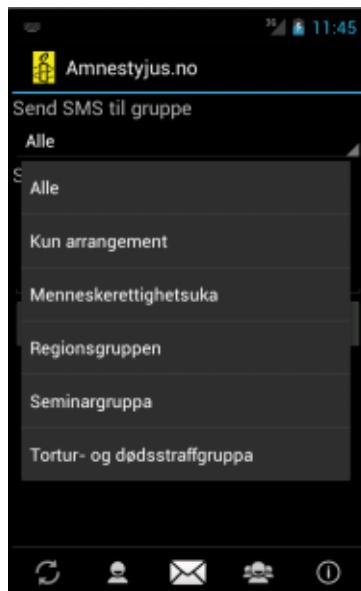
Brukeren av applikasjonen har mulighet til å sende ut tekstmelding til alle eller de forskjellige gruppene. Det velges gruppe via en spinner, som er en nedtrekksmeny. Hvilke verdier som skal vises i spinneren blir satt i et arrayadapter.

```
ArrayAdapter <CharSequence> adapter =  
    new ArrayAdapter <CharSequence>(this, android.R.layout.simple_spinner_item);  
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
```

For å få navnene på de forskjellige gruppene, blir det gjort et kall til DBAdapter sin metode finngrupper(), og alle navnene blir hentet ut via cursor.

Valget som brukeren velger vil så gå gjennom en ny cursor som sammenlikner navnet på gruppen, med de radene i tabellen medlemgrupper som inneholder gruppenavnet.

De radene som passer til kravet, vil hentes ut ifra metoden sendSMS(), det vil også bli sjekket i denne metoden om medlemmet har takket ja til å motta SMS fra Amnesty Jus Oslo.



Oversikt over mappen res 3

```

Cursor cur = db.finnGrupper();
if(cur.moveToFirst()){
    do{
        adapter.add(cur.getString(0));
        final Spinner s = (Spinner) findViewById(R.id.id);
        s.setAdapter(adapter);
        s.setOnItemSelectedListener(new OnItemSelectedListener() {

            public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2,
            long arg3) {
                int index = s.getSelectedItemPosition();
                final String gruppe = s.getItemAtPosition(index).toString();
                final Cursor cur2 = db.sendSMS ();
                btn.setOnClickListener(new Button.OnClickListener(){
                    public void onClick(View v){
                        if(cur2.moveToFirst()){
                            do{
                                String tableGruppe = cur2.getString(0);

                                if(gruppe.equals(tableGruppe)){
                                    String GSM = cur2.getString(3);
                                    String melding =txtMessage.getText().toString();

                                    if(melding.length()>0)
                                        sendSMS(GSM, melding);
                                    else{
                                        Toast.makeText(getApplicationContext(),
                                            "Vennligst skriv inn tekstmelding.",
                                            Toast.LENGTH_SHORT).show();
                                    }
                                }
                            }while(cur2.moveToNext());
                        }
                    }
                });
            }
        });
    }while(cur.moveToNext());
}
}

public void onNothingSelected(AdapterView<?> arg0) {
}

});
}while(cur.moveToNext());
}

```

4.3.8 Alle medlemmer

For å liste alle medlemmene er det brukt to klasser. Den første klassen ListAlleMedlemmer viser alle medlemmernes fornavn og etternavn. Det er mulig for brukeren å velge et enkelt medlem og få vist all informasjon som er lagret i databasen om det enkelte medlemmet. Her blir layouten vist dynamisk og ikke via en XML fil. Layouthen blir satt direkte i klassen, og det blir laget knapper ut ifra hvor mange linjer det er i den valgte tabellen. Siden denne fremvisningen inneholder flere medlemmer enn hva som kan vises på en side, er det lagt inn ScrollView, dette gjør det slik at informasjonen som vises kan gå utover størrelsen på den fysiske skjermen, og brukeren har mulighet til å bla nedover.



Oversikt over mappen res 4

```

ScrollView sv = new ScrollView(this);
    LinearLayout ll = new LinearLayout(this);
    ll.setOrientation(LinearLayout.VERTICAL);
    sv.addView(ll);

Alle medlemmene blir hentet ut ifra metoden finnAlle(), som ligger i DBAdapter. Her
brukes igjen Cursor og de to første strengene på hver rad blir hentet ut og vist som
knapper.

Cursor cur = db.finnAlle();

if(cur.moveToFirst()){
    do{
        Button btn = new Button(this);
        btn.setText(cur.getString(1)+" "+cur.getString(2));
        ll.addView(btn);
        final String brukerID = cur.getString(0);
        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                Bundle bundle = new Bundle();
                bundle.putString("brukerID", brukerID);
                Intent intent=new Intent(view.getContext(), HentMedlem.class);

                intent.putExtras(bundle);
                startActivityForResult(intent, 0);
            }
        });
    });

}while(cur.moveToNext());

```

Fremvisningen av all informasjonen blir gjort i klassen HentMedlem, her blir informasjonen hentet ut fra databasen på samme måte som i den første klassen, via en cursor, men forskjellen her er at denne klassen har en ekstern XML fil for layout. Her ligger det et felt textView for hver rad som skal vises.

```

<TextView
    android:id="@+id/visBrukerID"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```



Figur: Se informasjon om en person

4.3.9 Velg gruppe

Brukeren av applikasjonen har mulighet til å velge en enkelt gruppe, dette brukes klassen FinnGruppe til. Gruppenavnene listes opp som knapper slik at det er mulig å velge en enkelt gruppe og få fremvist alle medlemmene som er medlem i denne gruppa.

Det blir gjort et kall på metoden finnGrupper() i DBAdapter, og alle gruppentitlene blir listet ut ifra Cursor.

```
Cursor cur = db.finnGrupper();
```



Figur: Liste grupper.

5. Konklusjon

Det ble i starten av prosjektfasen satt en kravspesifikasjon i samarbeid mellom vår oppdragsgiver og gruppen. Vi har delt opp de kravene som ble satt i to deler, i tillegg til å legge til en tredje del for Android applikasjonen. Vi har revidert kravspesifikasjonen litt underveis, men føler de fleste kravene som ble satt i startfasen av prosjektet er oppnådd. Kravene som ble satt for hele webløsningen føler vi at har blitt løst på en tilfredsstillende måte, og at vi har lært nye måter å løse problemer innen webprogrammering på. Kravene innen Android applikasjonen var det for det meste vi i gruppen som satt, siden dette var en del vi selv ønsket å ha med for å få et bredere omfang på hovedoppgaven.

Android delen er nok delen som har vært mest krevende for oss, siden dette var helt ukjent for oss før prosjektstart. Vi føler vi har klart å få til det grunnleggende med en mobilapplikasjon, og er veldig fornøyd med forståelsen vi har fått. Applikasjonen har blitt et enkel og brukervennlig administrasjonsverktøy som har tilfredsstilt de kravene vi satt i starten av prosjektet.

Medlemssiden og administrasjons verktøy er et resultat på et godt prosjektarbeid synes vi. Vi har likt utviklingsmodellen vi har brukt, som ga oss muligheten til å revidere underveis ettersom vi så løsninger som var bedre og småfeil vi fant i koden. Vi ser på systemet som en brukervennlig og enkel side, som har kun sine nødvendige funksjoner. Det er lett forståelig å skjønne for studenten hvordan de skal melde seg inn, redigere brukeren eller melde seg ut hvis det er ønskelig. Medlemmene i styret kan oversiktlig skrive informasjons e-post til medlemmene, eller bruke de andre verktøyene på administratorsiden. Ut ifra det bruksområdet Amnesty Jus styret ønsket seg føler vi det har blitt et godt administrasjonsverktøy.

Grunnet manglende tilbakemelding fra oppdragsgiver i sluttperioden av prosjektet, har vi ikke fått lagt ut linker på hovedsiden til Amnestyjus.no, vi følte det var noe vi måtte få tillatelse fra kontaktperson før vi utførte. Dette er noe vi selvfølgelig kommer til å gjøre så fort vi hører fra dem.

Prosessrapport

Administrasjonsverktøy for Amnesty Jus Oslo

Margit Cecilie Haugen - s163289

Tonje Henriksen - s156049

Pernille Mohn - s163300

30.05.2012

Sammendrag

Dette prosjektet startet når vi fant oppgaven for Amnesty Jus Oslo på Høgskolen i Oslo og Akershus hovedoppgaveside. Etter å ha fått godkjente fra Amnesty Jus at de ville ha oss til å utføre deres oppgaver, så fikk vi tildelt veileder. Relativ fort etter dette kom i gang med hovedprosjektet.

Vi har gjennom prosjekttiden jobbet i 2 ukers bolker, som vi har delt inn i våre egne type faser. Hvordan samarbeidet og utviklingsmetoden vår har fungert kan det leses mer om i prosessrapporten.

Innholdsfortegnelse

Prosessrapport.....	78
Sammendrag	79
1. Innledning.....	80
2. Om gruppen	80
3. Om bedriften.....	80
4. Mål for prosjektet	80
5. Planlegging og metode	81
5.1. Utviklingsmodell	80
5.2. Overordnet ukesplan.....	82
5.3. Verktøy.....	83
Programvare.....	83
7. Samarbeid.....	83
8. Veiledning.....	84
8.1. Intern veileder.....	84
8.2. Ekstern veileder	84
9. Konklusjon.....	84

1. Innledning

Denne rapporten er prosessen delen til prosjektet, her er det skrevet om oss og om oppdragsgiveren. Om hvordan vi har arbeidet i prosjektet og hvilke mål vi hadde for prosjektet.

2. Om gruppen

Gruppen bak prosjektet består av Pernille Mohn, Tonje Henriksen og Margit Cecilie Haugen, vi går alle på Anvendt Data teknologi ved Høgskolen i Oslo og Akershus. Vi har jobbet med et administrasjonsverktøy for Amnesty Jus Oslo. Prosjektet startet med at vi fant oppdragsgiver gjennom Høyskolen i Oslo og Akershus, etter at Amnesty Jus hadde kontaktet skolen etter ønske om at noen ville ta dette oppdraget.

Vi har ikke tidligere jobbet sammen som en gruppe, men klarte tidlig å plassere våre oppgaver i gruppen, og utnyttet våre sterke sider. Vi føler vi har fått brukt mye av kunnskapene vi har lært ved vår 3.årlige utdannelse, men også fått utfordringer med å måtte videreutvikle våre data og prosjekt kunnskaper.

3. Om bedriften

Amnesty Jus Oslo er et student nettverk ved Universitet i Oslo det juridiske fakultet. Det er et studentnettverk som ble opprettet for å støtte og bekjempe menneskerettigheter. Nettverket består av flere undergrupper for hvilke interesse områder de jobber med. Til dags dato er det fire undergrupper i nettverket, Regionsgruppen, Tortur- og dødsstraffgruppa, Seminargruppa, Menneskerettighetsuka. Ellers jobber Amnesty Jus innenfor rammer gitt av Amnesty Norge etter behov og lyst.

I starten av prosjektet het bedriften Amnesty International Juridiske studentnettverk (AJIS), men byttet navn over nyåret til Amnesty Jus Oslo.

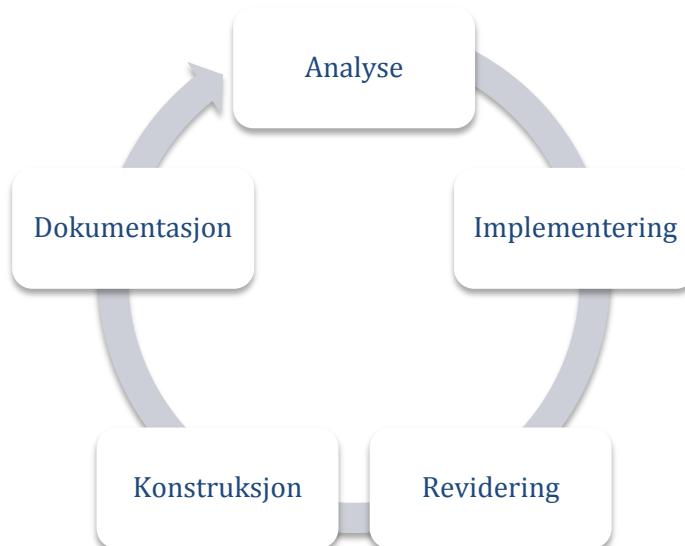
4. Mål for prosjektet

- Mål for prosjektgruppen.
- 1. Utvikle en brukervennlig og god side for Amnesty jus.
 - a. Å utvikle medlemssystem i henhold til kravspeken.
 - b. Utvikle et system som lever opp til oppdragsgiver forventninger.
- 2. Bruke fagkunnskapene våre etter 3 årlig Anvendt Data teknologi ved HIOA.
 - a. Fordype våre programmerings språk kunnskaper.
 - b. Fordype våre prosjektarbeid kunnskaper.
- 3. Lære oss grunnleggende Android mobilprogrammering.

5. Planlegging og metode

5.1. Utviklingsmodell

Fossefall- modellen er en systemutviklingsmodell, der aktivitetene blir inndelt i faser, der man gjør fase for fase, man skal ikke gå videre til neste fase før nåværende fase er ferdiggjort. Fossefall- metoden pleier å ha noen faser og noen gjennomgående aktiviteter. Vi har valgt å utvikle litt vår egen utviklingsmetode basert på grunnlaget av fossefalls metoden. Der vi har delt opp i to ukers faser der vi har små fossefall metoder. Vi synes slev dette var en bra metode siden en av fossefalls negative sider er at man ikke kan få tilbake til siste fase. I metoden har vi flere små underfaser , som kommer igjen i hver fase.



Figur: En fase i vår utviklingsmetode

I underfasene våre har vi tatt med Analyse, Implementering, Revidering, Konstruksjon og dokumentasjon. Med dette føler vi hele tiden at vi har hatt kontroll over de forskjellige fasene i prosjektet og at små feil vi fant ut underveis ble rettet opp der og da. Vi har hatt fokus på ha en god dialog innad i gruppen og med veileder, slik at fasene har fungert i forhold til prosjektet.

5.2. Overordnet ukesplan

Vi satte opp en ukeplan som gir en overordnet oversikt over hovedpunktene i de forskjellige fasene og hvem som har hovedansvaret for disse punktene.

Fase Nr:	Uke:	Pernille	Margit	Tonje
1	1-2	Foranalyse	Foranalyse	Foranalyse
2	3-4	Databasen	Databasen	Databasen
3	5-6	Medlemssiden	Dokumenter	Admin.side
4	7-8	Android/ Medlemssiden	Medlemssiden	Admin.side
5	9-10	Android/ Medlemssiden	Medlemssiden	Admin.side
6	11-12	Android/ Medlemssiden	Medlemssiden / Dokumentasjon	Admin.side
7	13-14	Android	Medlemssiden / Dokumentasjon	Admin.side
8	15-16	Android	Dokumentasjon	Admin.side
9	17-18	Android	Dokumentasjon	Admin.side
10	19-20	Android / Dokumentasjon	Dokumentasjon	Dokumentasjon
11	21-22	Dokumentasjon	Dokumentasjon	Dokumentasjon

Figur: ukeoversikt over prosjektperioden.

Vi begynte først med en kort analyse hvor vi hadde noen diskusjoner om hvordan vi ville at de forskjellige delene skulle se ut og hvordan vi skulle implementere de ulike funksjonene. Vi begynte deretter tidlig med programmeringen ettersom vi følte at vi hadde en god del jobb foran oss og jobbet samtidig parallelt med litt dokumentasjon. Vi konsentrerte oss først om medlemssiden og administrasjonssiden siden dette var det oppgaven gikk ut. Vi begynte deretter å sette oss mer og mer inn i Android. Tilslutt ble det bare sluttdokumentasjonen det gikk i.

Vi hadde god kommunikasjon med oppdragsgiver i begynnelsen av prosjektet med to møter hvor vi fikk oversikt over deres ønsker og behov og gikk gjennom noen ideer vi hadde for å gjøre oppgaven litt bredere. Etter vårt andre møte begynte vi for fullt med programmeringen av de forskjellige sidene. I sluttenten av prosjektet har vi dessverre ikke fått kontakt med oppdragsgiver for å høre om de er fornøyde med resultatet eller om de vil ha noen endringer ettersom dette ble midt i eksamensperioden deres.

Våre faste møter en gang i uken sørget for at prosjektet fikk en jevn fremdrift og at vi ikke behøvde å ta et skikkelig skippertak på slutten av prosjektet for å få alt ferdig.

5.3. Verktøy

Programvare

Eclipse Indigo Service release 2 – utvikling av Android mobilapplikasjon

Aptana studio 3 – PHP, HTML, JavaScript, utvikling av administrasjonsløsning på web

PHP Designer 2007 – Personal - PHP, HTML, JavaScript, CSS utvikling av administrasjonsløsning på web

Android virtual device manager – virtuell maskin for Android applikasjon

FileZilla – Tilkobling til FTP

WinSCP – tilkobling til FTP

Word – Alle rapporter

ArgoUML - utvikling av Use Case

6. Utviklingen av Applikasjonen

Vi har ingen tidligere erfaring med utvikling innen Java eller Android, så en stor del av utviklingen av applikasjonen har vært å sette seg inn i programmeringsspråket Java. Det er ulikt fra webprogrammering som vier bedre kjent med fra tidligere. Ved begynnelsen av utviklingen av applikasjonen brukte vi derfor en del tid på å sette oss inn i oppsettet av Android applikasjoner, hvilke deler som måtte være med og hvordan de forskjellige delene henger sammen.

Vi føler at vi har fått et godt innblikk i utvikling av Android applikasjoner og fått en bedre forståelse for programmering i språket Java.

7. Samarbeid

Vi har som sagt ikke jobbet sammen i gruppe tidligere, men vi ble fort kjent med hverandre og samarbeidet har fungert bra. Vi har jobbet en god del individuelt med forskjellige oppgaver, men vi føler likevel at alle har fått tatt del i alle deler av prosjektet.

Vi startet prosjektarbeidet med nesten daglige møter på skolen, men fant raskt ut når vi begynte med selve kodingen at det var mer effektivt å sitte hjemme hver for oss. Vi gikk etter hvert helt over til dette med et ukentlig møte på mandager hvor vi gikk gjennom

det vi hadde gjort og hva vi skulle gjøre til neste gang. For å kommunisere med hverandre når vi jobbet hjemme brukte vi MSN Messenger, telefon og e-post. Satt vi lenge fast på et problem spurte vi de andre på gruppen om de kunne se litt på det eller om de hadde noen input på hvordan det kunne løses. På denne måten fikk vi et godt samarbeid med få diskusjoner.

Vi er godt fornøyde med hva hver enkelt på gruppen har bidratt med og prosjektleder har vært full av energi og oppmuntring noe som har gitt oss mer motivasjon.

8. Veiledning

8.1. Intern veileder

Høgskolelektor Torunn Gjester har vært vår interne veileder fra Høgskolen i Oslo og Akershus. Vi har hatt ukentlige møter hver mandag, hvor vi har gått gjennom hva vi har gjort den siste uken, hva vi burde fokusere på videre fremover og fått tilbakemelding på hvordan vi ligger an i forhold til prosjektprosessen. Det har vært veldig bra å ha hyppige møter, så vi har fått god veiledning gjennom hele prosjektperioden.

8.2. Ekstern veileder

Alf Skre har fungert som vår kontaktperson fra Amnesty Jus Oslo. Ettersom dette ikke er deres kunnskapsfelt har vi ikke fått noe veiledning fra Amnesty Jus Oslo bortsett fra møtene vi har hatt hvor de har kommet med krav og ønsker.

9. Konklusjon

Til å begynne med var oppgaven vår bare å lage en medlemsside og en administrator side, men ettersom vi syntes dette var litt for snevert bestemte vi oss for å lage en mobilapplikasjon i tillegg, noe oppdragsgiver selv også likte ideen på. Denne avgjørelsen er vi veldig fornøyd med den dag i dag for vi føler selv at vi har lært veldig mye på denne oppgaven. Å lage en mobilapplikasjon er noe vi aldri har vært borte i før og det gikk derfor en god del tid til å sette seg inn i dette. Dette ser vi egentlig bare på som positivt siden vi har lært desto mere på grunn av det. Ved å lage en mobilapplikasjon har vi blant annet lært hvordan java fungerer og om hvordan klasser og metoder henger sammen. Vi har også fått satt oss mer inn i hvordan PHP fungerer og også lært oss litt javascript. Det har dessuten vært veldig lærerikt å få ha en virkelig oppdragsgiver som er avhengig av at vi gjør en god jobb. Dette har fått oss til å gi litt ekstra med tanke på at det er noen som faktisk skal bruke det.

Vi mistet litt kontakten med oppdragsgiver på slutten av prosjektet og vi har derfor ikke fått tilbakemeldinger fra dem på om de er fornøyde med resultatet eller ikke. Dette er selvsagt en stor ulempe, men vi synes vi har gjort det vi kan for å få kontakt med vår kontaktperson og at det ikke er noe vi kunne ha gjort annerledes der. Vi føler oss derfor fornøyde med egen innsats og med resultatet.

6. Kilder

Internett

Table row like a link (22.09.09) URL: <http://stackoverflow.com/questions/1460958/html-table-row-like-a-link> [Lest: 29.03.12]

Digg-style Pagination (2006) URL:

<http://www.strangerstudios.com/sandbox/pagination/diggstyle.php> [Lest: 07.04.12]

PHP: Exporting Data to Excel (06. 01.06) URL: <http://www.the-art-of-web.com/php/dataexport/> [Lest: 09.04.12]

Form Handling with PHP Lesson 5 - Attachments (15.06.10) URL:

<http://www.apptools.com/phptools/forms/forms5.php> [Lest: 14.04.12]

CSS For Bar Graphs (29.06.09) URL: <http://www.cssbakery.com/2009/06/data-visualisation-flexible-bar-graphs.html> [Lest: 21.04.12]

Brukst som oppslagsverk gjennom hele perioden:

URL: <http://snl.no/>

URL: <http://www.W3.org>

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

URL: <http://developer.android.com>

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

Bøker

Gargenta, Marko (2011) *Learning Android*. Sted: O'Reilly Media, Inc.

Kristoffersen, Bjørn (2009) *Databassystemer 2. utgave*. Sted: Univeritetsforlaget

Hasle, Thor E. (2008) *Systemutvikling - Applikasjoner og databaser*. Sted: Cappelen

7. Ordliste

A

Array – se Matrise

B

Boolean – returnerer verdi som sann eller usann(True/false)

C

Cookie – datamengde som en nettleser mottar fra bruker, lagres som fil på

D

Datatype – hvilke type data som blir lagret i en gitt variabel

Div-tag – en spesifikk seksjon i et dokument

Dynamisk – i bevegelse, endres

G

Grensesnitt – kommunikasjon mellom systemer

I

Input – begrep som omtegner noe som blir satt inn

If-setning – gir en betingelse for å kunne utføre handlig.

int – tallverdier

L

Layout – viser til hvordan sider/skjermbilder skal fremvises

M

Matrise – eng. Array. Tabeller eller flerdimensjonale variabler

MIME – Internet standard som viderefører formatet til en epost.

Metode – definerer instruksjonene i en klasse

O

Output – begrep som omtegner noe som kommer ut og blir vist.

Objektorientert programmering – programmering som støtter seg på objekter fremfor prosedyrer.

P

Pixel – det minste elementet i et digitalt bilde

R

Rammeverk – objekter som kan utveksles mellom programmer over internett, uavhengig av programmeringsspråk

S

Session – Variabler som blir lagret om en enkelt bruker, så lenge nettsiden er aktiv for bruker.

SMS – Short Message Service

String – sekvenser av tekst eller tall

V

Validering – bli gyldig, sjekker om en verdi som er satt er gyldig.
brukerens datamaskin

Variabel – Betegnelse for et vilkårlig element

W

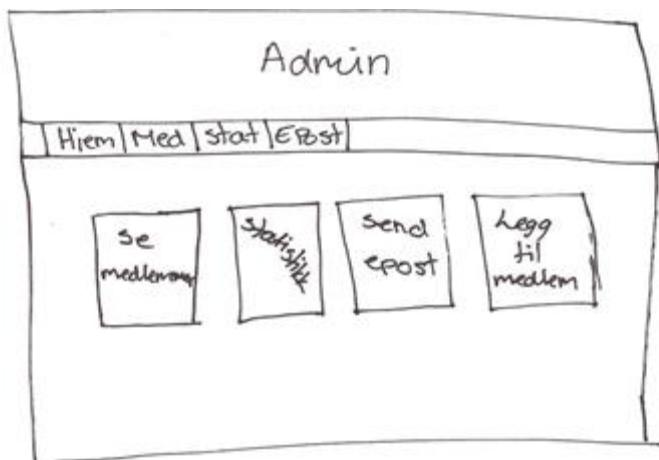
While-løkke – Utfører en handling et spesifikt antall ganger.

Å

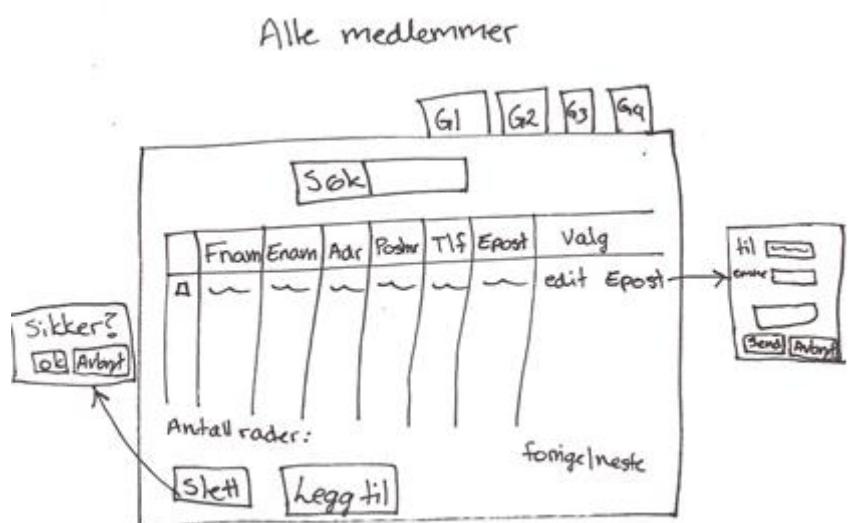
Åpen kildekode – programvare distributert under forutsetning av at kildekoden skal gjøres tilgjengelig for brukerne

8. Vedlegg

9.1 Skisser



Skisse 1 - Admin forsiden



Skisse 2 - Admin vis medlemmer 1

Statistikk

Anl. med. :	11
G1 :	2
G2 :	6
G3 :	0
G4 :	3

De fem nyeste

Fnamn	Enamn	Dato

Jan	Feb	Mars	April	Mai	Juni	Juli	Aug	Sept
1	1	1	1	1	1	1	1	1

Jan	Feb	Mars	April	Mai	Juni	Juli
1	1	1	1	1	1	1

Skisse 3 - Admin statistikk 1

Registrering

Fornavn :

Ettarnavn :

Adresse :

Postnr :

Stad : autz

Tlf :

Epost :

Passord :

Avgangssår

kryss av for gruppe

- G1
- G2
- G3
- G4

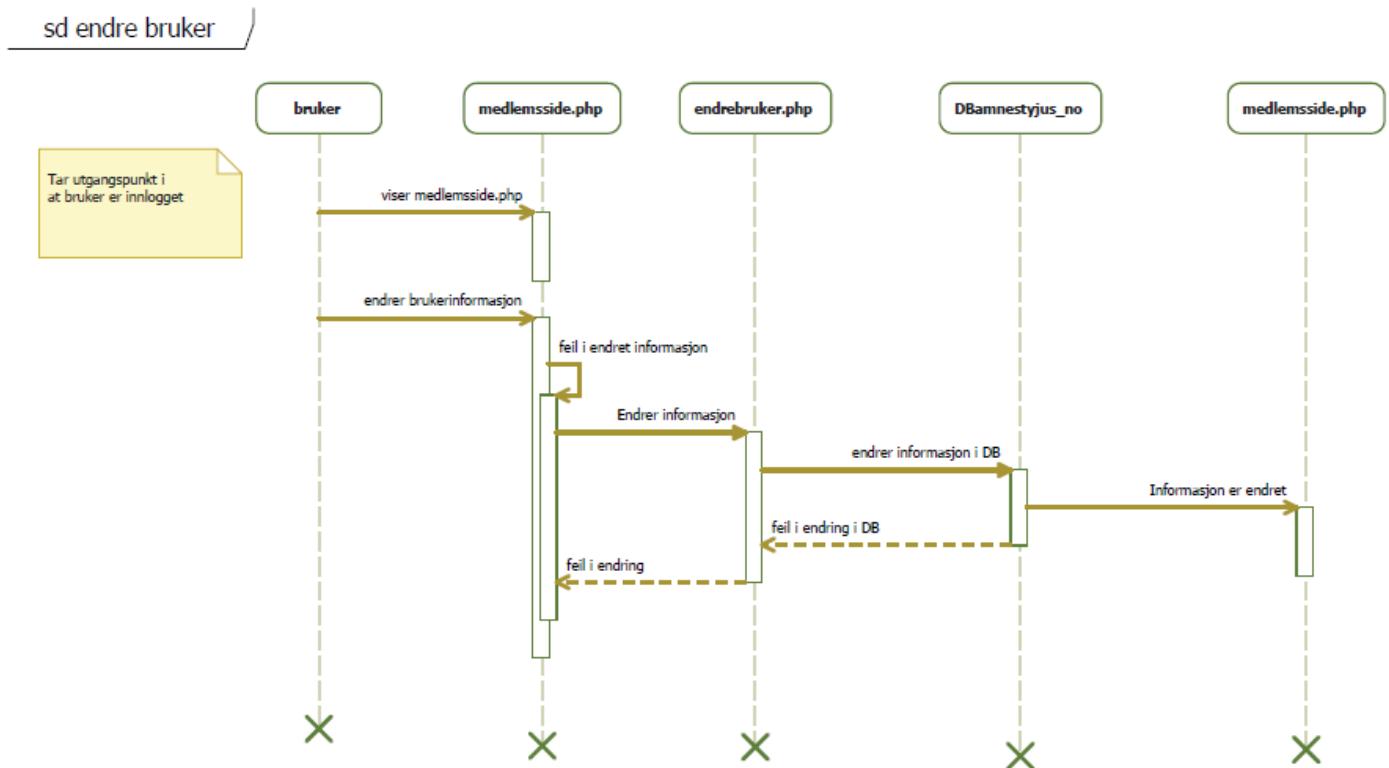
Tidligere erfaring

Jeg ønsker ikke sms

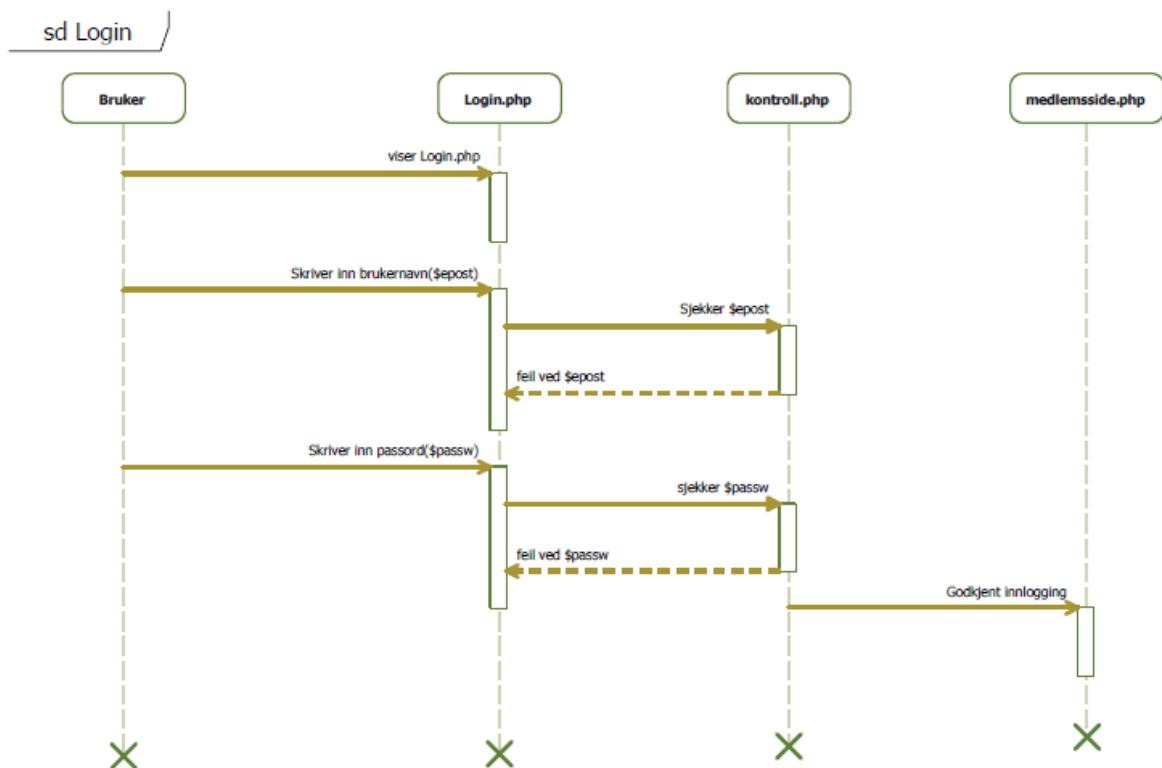
Register

Skisse 4 - Registrering medlem 1

9.2 Sekvensdiagram

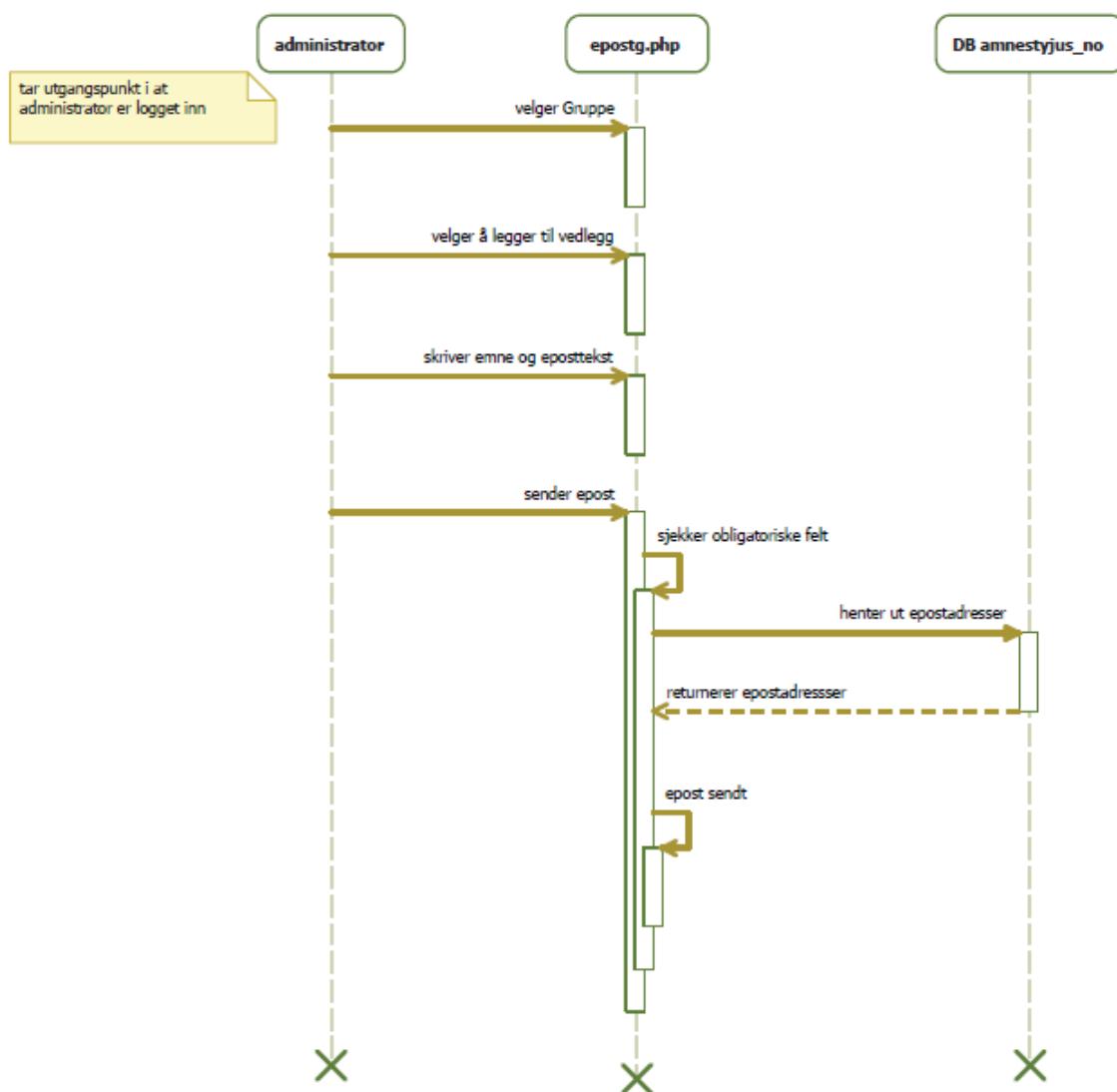


Sekvensdiagram 1 - endre bruker 1



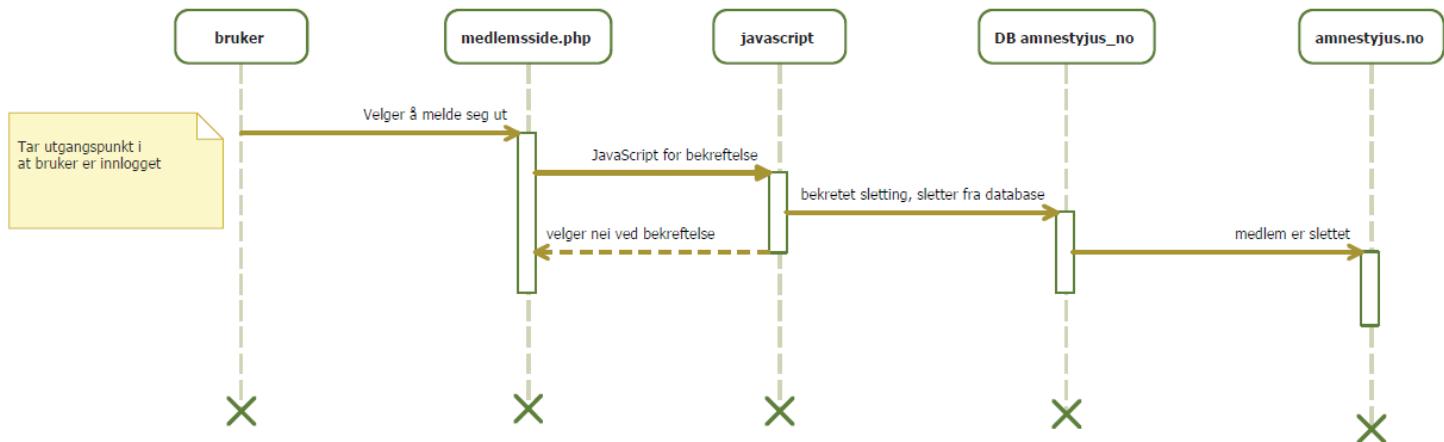
Sekvensdiagram 2 - Bruker login 1

sd sende post



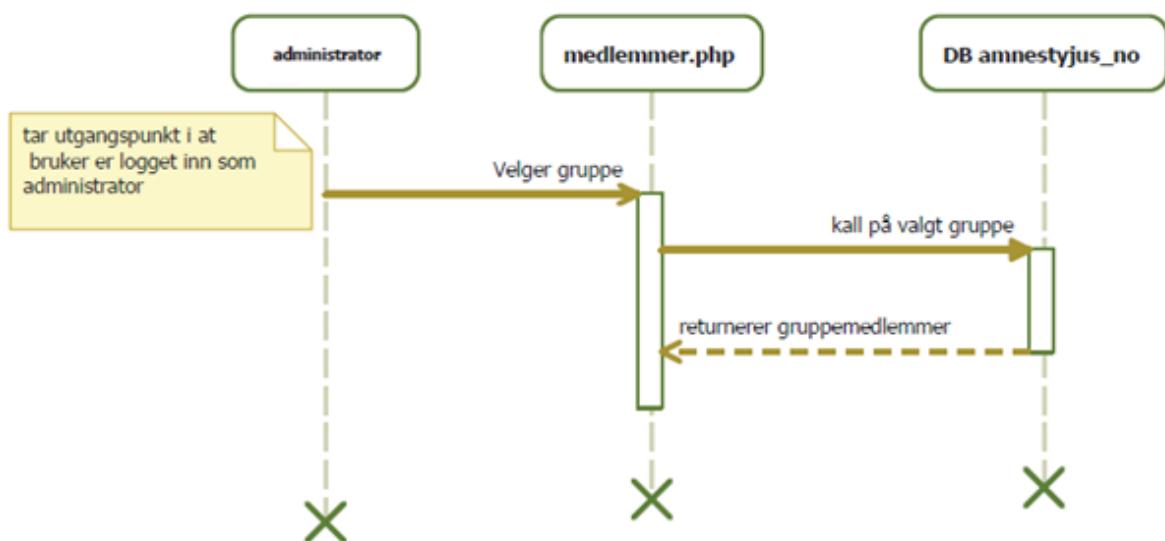
Sekvensdiagram 3 - Admin sende epost 1

sd Slette bruker



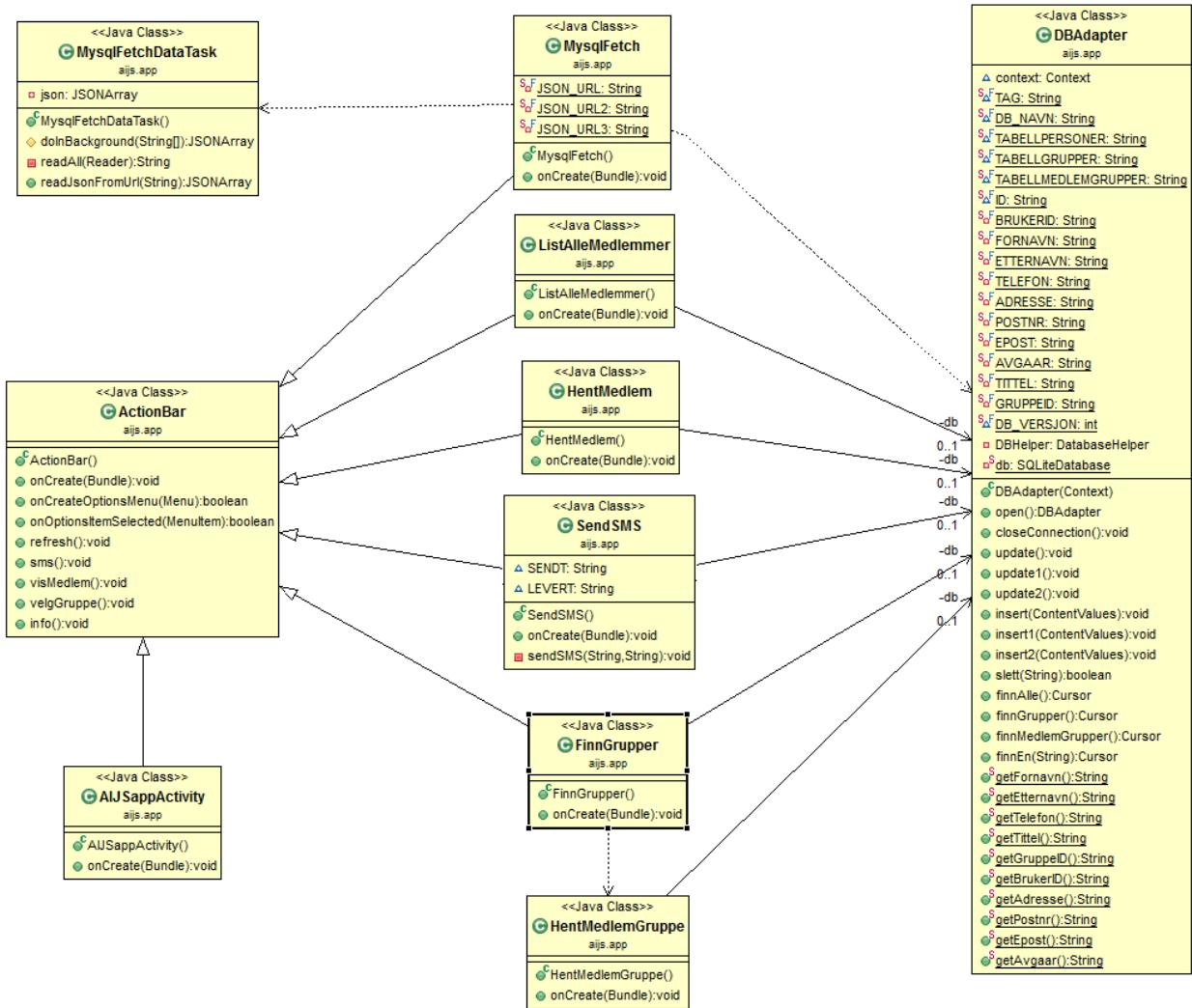
Sekvensdiagram 4 - Bruker sletter seg 1

sd viser gruppemedlem



Sekvensdiagram 5 - Admin viser grupper 1

9.3 Klassediagram av applikasjon



Klassediagram 1 - til mobilapplikasjonen