

# Refaktorering af en statisk webseite til en dynamisk webseite

---

*Afgangsprojekt udført ved Danmarks Tekniske Universitet*

**Forfatter:** Ahmet Yildirim  
**Vejleder:** Finn Gustafsson  
**Afleveringsdato:** 27. maj 2013



Danmarks Tekniske Universitet

**Forfatter:** Ahmet Yildirim - s093468

**Dansk titel:** Refaktorering af en statisk webside til en dynamisk webside

**Løbetitel:** IMM-B.eng-2013-8

**English title:** Refactoring of a static webpage to a dynamic webpage

**Projektperiode:** 18. februar 2013 – 27. maj 2013

**Omfang:** 20 ECTS-point

**Uddannelsessted:** Danmarks Tekniske Universitet

**Institut:** Institut for matematik og computer science

**Vejleder:** Finn Gustafsson

**Ahmet Yildirim**

---

# 1. Forord

Denne rapport er udarbejdet som et afgangsprøveprojekt på IT-diplomingeniøruddannelsen for Institut for matematisk modellering ved Danmarks Tekniske Universitet. Projektet har vægtningen 20 ECTS-point og er udarbejdet i perioden 18. februar 2013 – 27. maj 2013.

Rapporten omhandler refaktorering af en statisk webside til en dynamisk webside i forbindelse med interessentskabet Rørvig Vandværk. I denne forbindelse er der lavet en komplet renovering og omstrukturering af en webside, [www.rorvigvand.net](http://www.rorvigvand.net).

Rapportens primære formål er at danne grundlag for evalueringen af den studerendes diplomingeniøruddannelse ved Danmarks Tekniske Universitet, og rapportens målgruppe er derfor primært den tilknyttede vejleder, censor samt personer med en generel forståelse af Informatik. Sekundært anses rapporten at have interesse for de personer, som skal anvende det udviklede produkt i forbindelse med driften af Rørvig Vandværk.

I forbindelse med rapportens tilblivelse skal der rettes stor tak til vejleder Finn Gustafsson, Institut for matematik og computer science DTU, for stor behjælpelighed og vejledning i forbindelse med materialet samt forståelse for, hvordan interessentskabet er sammensat.

Derudover rettes der stor tak til Finn Gustafsson for konstruktiv vejledning og tålmodighed under projektførelsen.

## 2. Abstract

Rørvig Water Company I/S was founded on 15 August 1950 as a private partnership, unsubsidized by either municipality or state, with the purpose of securing the drinking water supply of three areas registered, mainly, as summer cottage areas: Rørvig, Nakke and Hov Vig.

Rørvig Water Company's Board of Directors consists of five members elected by the representatives, whether from within or from outside their circle. The representatives, forming the supreme authority, number a total of twenty one members and nine deputy members.

Of the members, six are permanent residents of the areas in question, and fifteen are summer cottage owners, whereas the deputy members are made up by three local residents and six summer cottage owners.

At the yearly shareholder meeting in April/May, elections are held for seven members and three deputy members.

Additionally, two employees are engaged at the water company, being responsible for the daily operations and transactions of affairs, respectively.

The purpose of this project was to create a website for Rørvig Water Company whereby people with no particular knowledge of IT may nevertheless be able to use the website and take advantage of the implemented module for information sharing, thereby managing data relating to directors of the Board, representatives and deputy members.

### 3. Indledning

Rørvig Vandværk I/S blev stiftet d. 15. august 1950 som et privat interessentselskab, som uden kommunal eller statslig støtte har til opgave at sørge for drikkevandsforsyningen i de tre kommunale matrikelområder Rørvig, Nakke og Hov Vig.

Rørvig Vandværks bestyrelse består af 5 medlemmer, som vælges af repræsentantskabet inden for eller uden for dettes kreds. Repræsentantskabet, som er selskabets højeste myndighed, består af 21 repræsentanter og 9 suppleanter.

Repræsentanterne består af 6 fastboende og 15 sommerhusejere, hvorimod suppleanterne består af 3 fastboende og 6 sommerhusejere.

På interessentmødet hvert år i april/maj foretages valg af 7 repræsentanter og 3 suppleanter.

Derudover er der på selve vandværket ansat 2 personer, som står for den daglige drift og forretningsførelsen.

Formålet med dette projekt er at sikre personer uden større forståelse for IT kan anvende denne webside og benytte sig af det implementerede modul til informationsdeling og administrere data, som vedrører medlemmer af bestyrelsen, repræsentanterne, og suppleanterne.

## 4. Problemformulering

Statiske websiders dage er ved at være talte, da der efterhånden skiftes over til dynamiske websider. På trods af det findes der stadig statiske websider, men da det efterhånden er blevet nemmere for websidebrugerne at tilgå dynamiske websider, eftersom der ikke behøves teknisk indsigt i web-programmering, er der flere og flere, som skifter over.

I dette eksamensprojekt ønskes der en refaktorering og renovering af hele hjemmesiden, således at dele af indholdet bliver dynamisk. Dette vil medføre en lettere måde at håndtere indholdet på websiden for de personer, som redigerer indholdet af den. Til dette ønskes en udskiftning af frames, som erstattes med CSS, omlægning af hele menuen, CRUD-implementering, login system-implementering, samt diverse tilpasninger. Alt i alt skal det ikke-statiske indhold kunne opdateres via et admin-login, som gennem PHP-scripting og MySQL backend tager hånd om opdateringer.

# Indholdsfortegnelse

1. Forord .....	3
2. Abstract .....	4
3. Indledning .....	5
4. Problemformulering .....	6
5. Metode .....	9
6. Værktøjer .....	10
7. Analyse .....	11
7.1. Formål .....	11
7.2. Mål .....	11
7.3. Målgruppe .....	11
7.4. Indhold .....	11
7.5. Funktionelle krav .....	12
7.6. Non-funktionelle krav .....	12
7.7. Database analyse .....	13
8. Design .....	14
8.1. Sitemap .....	14
Wireframes .....	16
8.2. Comps .....	16
8.3. Arkitektur .....	17
8.4. Databasesdesign .....	18
9. Implementering .....	23
9.1. Use cases .....	23
9.2. Flowchart .....	29
9.3. Sekvensdiagram .....	35
9.4. Frontend/Backend .....	39
9.5. Input validering .....	57
9.6. Sikkerhed .....	60
10. Test .....	65
10.1. HTML validering .....	65
10.2. CSS validering .....	67

10.3. Browser test .....	68
10.4. Whitebox test .....	69
10.5. Blackbox test.....	70
10.6. Konklusion på test .....	79
11. Brugervejledning .....	80
12. Projektplan .....	93
13. Installation .....	94
14. Diskussion .....	95
14.1. Forbedringstiltag .....	95
14.2. Metode .....	95
14.3. Vedligeholdelse .....	96
15. Konklusion .....	97
16. Figurbetegnelse .....	98
17. Tabelbetegnelse .....	103
18. Litteraturliste .....	105
18.1. Bøger .....	105
18.2. Internet.....	106
19. Bilag .....	107
19.1. Wireframes.....	107
19.2. Comps .....	113
19.3. Use case diagrammer .....	114
19.4. Flowchart diagrammer .....	116
19.5. Sekvensdiagrammer .....	124
19.6. Blackbox test.....	129
19.7. Browsertest .....	159



## 5. Metode

Vandfaldsmodellen<sup>1</sup> er en softwareudviklingsmodel, og er benyttet i forbindelse med dette projektforsløb. Vandfaldsmodellen indeholder faserne: kravspecifikation, design, implementation, integration, test/afprøvning, installation og vedligeholdelse.

1. Kravspecifikation – beskrivelse i ord hvad kravene er.
2. Design – diagrammer og figurer, som danner grundlag for hvordan udvikleren skal udvikle selve systemet.
3. Implementation – systemudvikleren implementerer selve systemet.
4. Integration – integration af de forskellige komponenter. Klientdelen integreres med serverdelen.
5. Test/afprøvning – udførelse af test og afprøvning af alle implementerede dele med formålet at finde fejl og mangler samt verificere, at den implementerede kode er korrekt.
6. Installation – systemet installeres på platformen og gøres køreklar.
7. Vedligeholdelse – introduktion til ny funktionalitet, og hvordan fejl kan fjernes.

Hver fase gennemløbes nedadgående, således at alle faserne slavisk gennemarbejdes fra 1-7. Først efter afslutning på en af faserne kan der bevæges videre til næste fase. Et af de vigtigste succeskriterier på projekter er at tiden overholdes, hvilket er med til at definere tid som en kritisk faktor for et projekt. Vandfaldsmodellen er en metode som er med til at gøre det lettere at overholde diverse deadlines, da hver fase kan anskues som en vigtig milestone for projektet.

Spiralmodellen<sup>2</sup> er også udnyttet i forbindelse med dette projektforsløb. Denne model er en kombination af iterativ systemudvikling og faserne fra vandfaldsmodellen. Dette betyder at der ved hver runde i spiralen, kan udgives en delmængde af produktet. Dette har været relevant, da projektforsløbet har bestået i små iterative forsløb (processer) med forskellige opgaver, hvor der efter hver runde i udviklingsforsløbet er blevet udgivet en delmængde af produktet. Denne model er også effektiv hvad angår risikostyring, da der i hver iteration planlægges hvorledes den næste iteration skal forsløbe.

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Waterfall\\_model](http://en.wikipedia.org/wiki/Waterfall_model)

<sup>2</sup> [http://en.wikipedia.org/wiki/Spiral\\_model](http://en.wikipedia.org/wiki/Spiral_model)

## 6. Værktøjer

I nedenstående liste optræder de værktøjer og teknologier/platforme som er anvendt i forbindelse med dette projekt:

Værktøjer/Programmer:

1. Visio 2013 – brugt til flowchart diagrammer, sekvensdiagrammer, use case diagrammer, wireframes og sitemap.
2. Netbeans – brugt som IDE (integrated development environment) til udvikling i PHP, html, css og javascript.
3. Phpmyadmin – brugt til at administrere MySql databaseserveren.
4. Filezilla – brugt som ftp klient til at uploade filer til webserveren.
5. Browsers – brugt som klient til systemet (Internet Explorer, google chrome, Mozilla Firefox og Apple Safari).
6. DBSchema – brugt til E/R diagrammer.
7. GanttProject – brugt til projektstyring.
8. Wampserver – brugt som lokal webserver, lokal mysql database og php oversætter.

Teknologier/Platforme:

1. HTML 4.01 transitional er brugt, da denne DTD indeholder alle HTML elementer og attributter, som inkluderer præsentationselementer og frarådede elementer.
2. PHP version 5.3.
3. MySql version 5.1
4. Apache version 2.2

## 7. Analyse

### 7.1. Formål

Rørvig vandværks websides primære formål er at dele informationer til de pågældende interessenter. I 2009 var der registreret 3907 interessenter, hvoraf ca. 700 eller 18 % af disse var helårsbeboere og de resterende var sommerhusbeboere. Websidens sekundære formål er at yde en service for de ansatte, og derved gøre det lettere at styre driften samt administrere den ledende organisation i selskabet.

### 7.2. Mål

Denne webside muliggør, at de ansatte ved hjælp af et CRUD-modul nemt kan håndtere data, som vedrører den ledende organisation i selskabet.

### 7.3. Målgruppe

Målgruppen for denne webside er personer, som har forbindelse til Rørvig Vandværk, herunder interessenter og personale. Kravet til disse er, at de er indforstået med hvordan organisationen fungerer rent praktisk.

### 7.4. Indhold

Målgruppen for denne webside vil som primært se nedenstående informationer om Rørvig vandværk:

- Historien bag Rørvig vandværk
- Advarsler
- Diverse analyseblade
- Takstblade
- Regulativer
- Vedtægter
- Meddelelser til interessenterne
- Informationer omkring personalet
- Informationer omkring webmasteren
- Informationer omkring repræsentanterne
- Informationer omkring suppleanterne
- Informationer omkring bestyrelsen

Sekundært vil indholdet af websiden bestå af et administrationsmodul, som vedrører de personer, som styrer driften af vandværket, og de personer, som sidder i de ledende organer. Målgruppen for dette administrationsmodul vil kunne se indhold svarende til informationer om personer i de ledende organer, herunder:

- Oprettelse af nye personer i de ledende organer
- Redigering af eksisterende personer i de ledende organer
- Sletning af eksisterende personer i de ledende organer
- Opdatering af data for personer i de ledende organer
- Tildeling af medlemskaber for personer i de ledende organer
- Fjernelse af medlemskaber for personer i de ledende organer

## 7.5. Funktionelle krav

Det er et krav fra kunden, at systemet skal udvikles i PHP, da man ønsker et open source system, hvilket er gratis. Til databasen ønskes der MySQL, der som minimum er på 3. normalform. Der findes tre forskellige typer brugere. Den første er almindelige brugere, som bruger siden til at indhente informationer. Den anden er administratorbrugere, som administrerer data om personer i de ledende organer. Den tredje er system administratorbrugere, som håndterer administratorbrugerne. Det er kun administratorbrugerne, der har adgang til administrationsmodulet. Systemkravet til administratormodulet er, at det skal være muligt at oprette, slette og ændre data i databasen (CRUD). Derudover skal alle html sider valideres via et online valideringsværktøj samt validering af alle input felter, hvor mangler/fejl i inputfeltet skal generere fejlmeddelelse til brugeren.

## 7.6. Non-funktionelle krav

### Usability

Det er ikke et krav, at brugeren er vant til at arbejde med webside, men det anbefales dog at have lidt erfaring med det.

### Reliability

Tab af data må ikke ske. Databasen bør have en backup, således at selskabet ikke mister sine data.

### Performance

Performance for systemet er svært at definere, da brugerne ikke anvender samme computere, dvs. computere med samme hardware og browsere.

Respons tiden for systemet er ikke kritisk lav, og svartider op til 5 sekunder accepteres. Denne respons tid er sat ud fra udviklerens egne erfaringer om maximal ventetid.

### Implementering

Under selve udviklingsmiljøet vil der gøres brug af en WAMP stak. Denne stak består af en MySQL database, apache webserver og PHP compiler. Denne WAMP stak kører på en Windows platform. Til udviklingsværktøjet vil der gøres bruges af Netbeans samt MySQL administrator. Det er et krav, at kodelister er veldokumenteret med kommentarer.

## Test

Det er et krav, at selve funktionaliteten er testet i gennem og fejlfri. Derudover kræves, at der er foretaget en browsertest af de mest anvendte browsere (Explorer, Chrome, Firefox og Safari). Der ønskes også, at der er foretaget en validering af grænsefladen vha. et godkendt valideringsværktøj (W3C validator).

## 7.7. Database analyse

### Kravspecifikation

Der er udarbejdet en kravspecifikation for at identificere reelle informationer om databasen. De mest essentielle informationer er således:

- Personer med navn, adresse, telefon, mobil, e-mail, sommerhusadresse, sommerhuspostnummer og sommerhustelefon.
- Repræsentanter med navn, adresse, telefon, mobil, e-mail, sommerhusadresse, sommerhuspostnummer, sommerhustelefon, repræsentantvalgår og suppleant til bestyrelsen.
- Suppleanter med navn, adresse, telefon, mobil, e-mail, sommerhusadresse, sommerhuspostnummer., sommerhustelefon, suppleantvalgår og suppleantprioritet.
- Billeder af personer.
- Administrative brugere med brugernavn, kodeord og e-mail.
- Tæller til at holde styr på websidens besøgende med tid og ip-adresse.

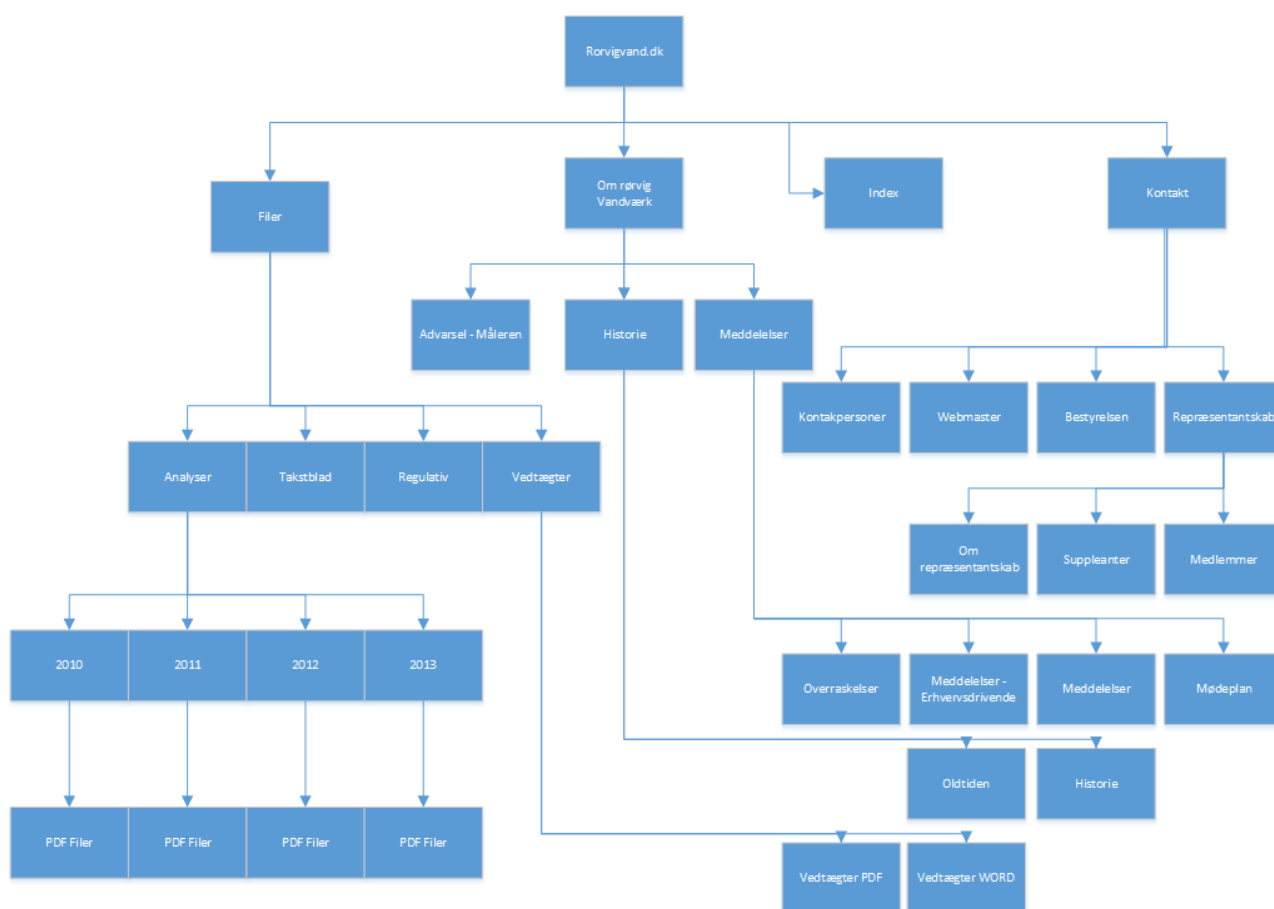
# 8. Design

## 8.1. Sitemap

Et sitemap er et diagram, som enten kan repræsenteres grafisk eller på XML format. Formålet med et sitemap er at vise brugerne den hierarkiske opsætning af websiden, således at brugerne nemmere kan navigere rundt på websiden og dermed finde indholdet på websiden. Et sitemap kan også benyttes af webdesigneren som et planlægningsværktøj til at give overblik over websiden, og hvordan den skal organiseres.

### Sitemap – Brugere uden login

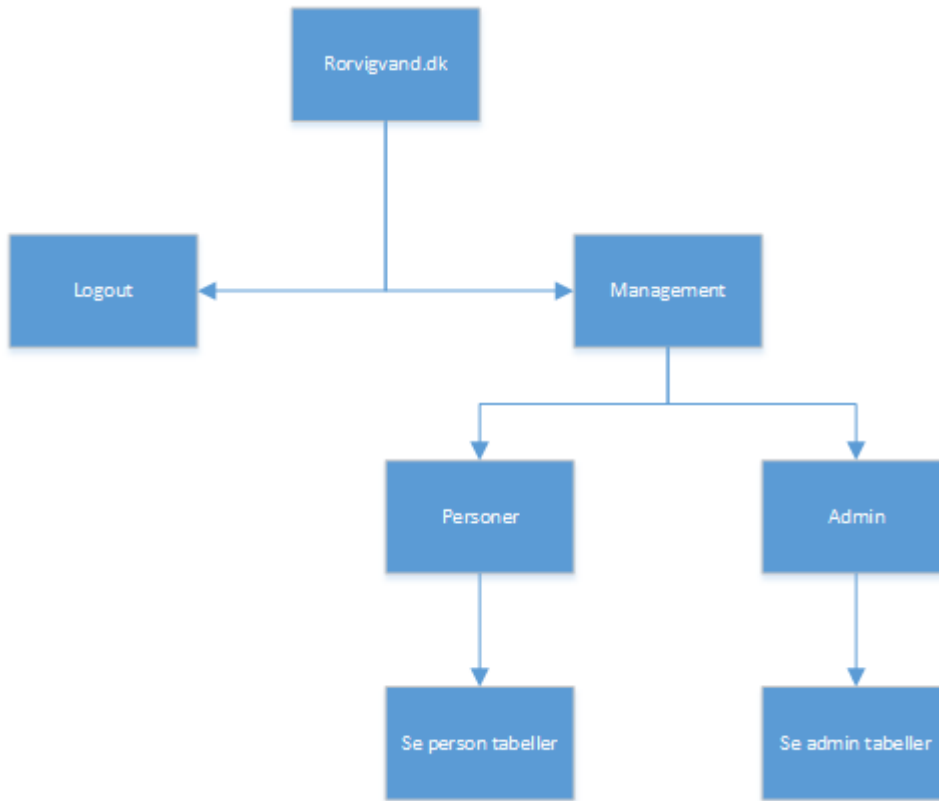
Nedenstående figur 1 visualiserer hvilke muligheder der er for brugere uden et login. Dette illustrerer samtidig også hvordan brugeren kan navigere rundt vha. menuen.



Figur 1 - Sitemap - Brugere Uden Login

## Sitemap – Brugere med login

Nedenstående figur 2 visualiserer, hvilke ekstra muligheder der er for brugere med et login. Dette illustrerer samtidig også hvordan brugeren kan navigere rundt vha. menuen. Dette sitemap indeholder kun de ekstra muligheder, brugere med login har.



Figur 2 - Sitemap - Brugere Med Login

## Wireframes

Wireframes er figurer/tegninger, som viser, hvorledes hver enkelt side vil fremstå, og giver et designmæssigt indblik i, hvordan den visuelle del af websiden kommer til at se ud.

Den visuelle del af websiden er designet vha. wireframes, som giver udvikleren et udgangspunkt/grundlag for, hvordan websiden skal implementeres, derfor er der designet wireframes for hver enkelt side. Der er lavet wireframes for hver enkelt side, som kan findes i bilagsafsnittet:

1. Index – figur 32
2. Meddelelser – figur 33
3. Mødeplan – figur 34
4. Historie – figur 35
5. Kontaktpersoner – figur 36
6. Webmaster – figur 37
7. Medlemmer – figur 38
8. Bestyrelsen – figur 39
9. Suppleanter – figur 40
10. Person administration – figur 41
11. Admin administration – figur 42
12. Login – figur 43
13. Glemmt login – figur 44

## 8.2. Comps

Der er anvendt CSS til at designe grænsefladen. Dette style sheet er inkluderet i HTML header delen på hver af websidens sider<sup>3</sup>.

### Baggrundsbillede

Der er anvendt et baggrundsbillede, som er gældende for alle websidens sider. Dette baggrundsbillede kan associeres med vand. Dette baggrundsbillede dækker hele siden.

Det anvendte baggrundsbillede kan ses i bilagsafsnittet, figur 45.

### Fonts

Der er anvendt sans-serif som teksttype for alle websidens tekster. Størrelsen på teksterne er forskellig alt efter overskrift, brødtekst og sidefoden. Tekstfarven er sort.

### Drop down-menu

Drop down-menuen udformes som en lineær gradient af to farver: #4f5964 og #5f6975. Dette betyder, at gradienten vil bevæge sig i en lige linje fra toppen mod bunden med en gradvis farveændring – fra den første til den anden farve – ned ad linjen.

---

<sup>3</sup> CSS filen kan findes på CD'en (css/style.css)



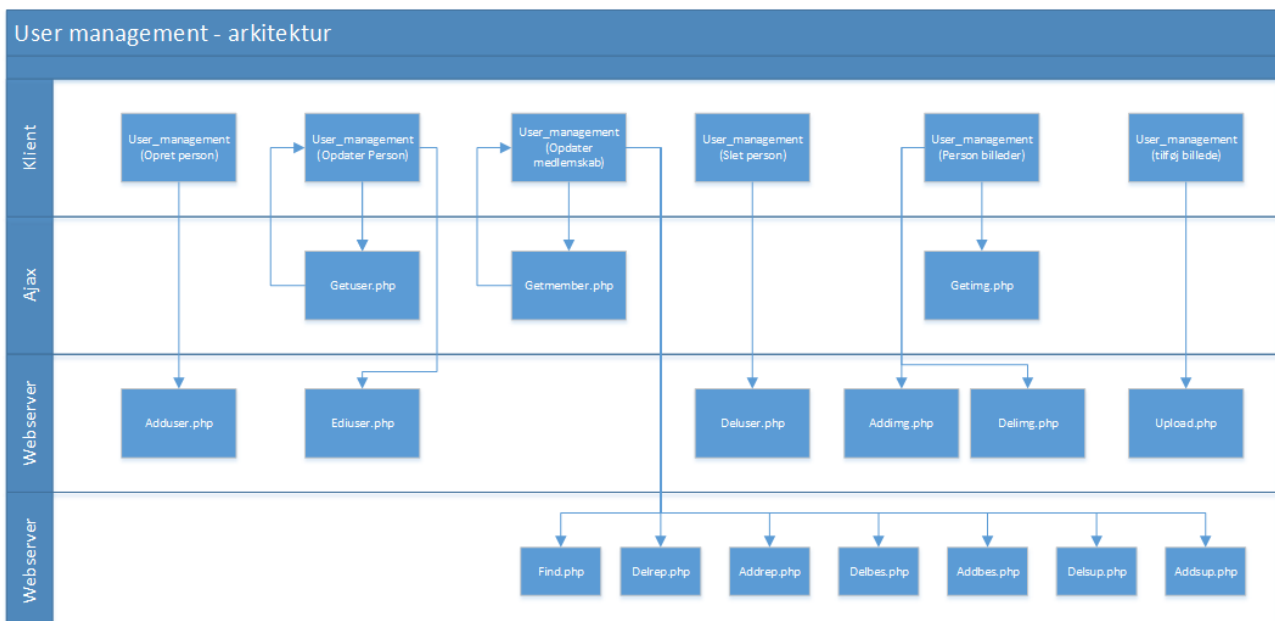
Denne lineære gradient supportes af de mest populære browsere: Internet Explorer 5.5+, Google Chrome 1+, Mozilla Firefox 3.6+, Apple Safari 4+, Android 2.3+ og Opera 11.1+.

### 8.3. Arkitektur

Den overordnede arkitektur kan opdeles i 3 dele. Klient (Browseren), Ajax (asynkron dataudveksling med webserver) samt webserveren. Klienten er der, hvor brugeren kan se de forskellige elementer samt indtaste data i de forskellige felter. Ajax benyttes til asynkron dataudveksling med serveren, således at brugeren får specifik HTML kode fra webserveren, baseret på det input, webserveren modtager. Webserveren er der, hvor selve kommunikationen med databasen foreligger.

Denne figur viser hvorledes sammenhængen mellem klienten, ajax og webserveren er.

1. User\_management (Opret person) – her sendes data direkte til webserveren (adduser.php)
2. User\_management (Opdater person) – her sker en udveksling af data med ajax (getuser.php), således at klienten får specifik HTML baseret på det input, der er sendt til getuser.php. Klienten har nu modtaget specifik HTML og kan sende data til webserveren eduser.php
3. User\_management (Opdater medlemskab) – her sker en udveksling af data med ajax (getmember.php), således at klienten får specifik HTML baseret på det input, der er sendt til getmember.php. Klienten har nu modtaget specifik HTML og kan sende data til webserveren (find.php/delrep.php/addrep.php/belbes.php/addbes.php/delsup.php/addsup.php)
4. User\_management (Slet person) – her sendes data direkte til webserveren (deluser.php)
5. User\_management (Person billeder) – her sker en udveksling af data med ajax (getimg.php), således at klienten får specifik HTML baseret på det input, der er sendt til getimg.php. Klienten har nu modtaget specifik HTML og kan sende data til webserveren (adding.php/delimg.php)
6. User\_management (Tilføj billede) – her sendes data direkte til webserveren (upload.php)



Figur 3 - Overordnet Arkitektur

## 8.4. Databasedesign

Når der designes en database for en virksomhed, er det vigtigste mål at skabe en nøjagtig gengivelse af de data, relationer mellem data og begrænsninger på de data, der er relevante for virksomheden. Normalisering er en database-designsteknik, der begynder ved at undersøge forholdet mellem de data, der er vigtige for virksomheden.

En E/R model er en top-down-tilgang til databasedesign, der indledningsvis identificerer, om de enheder og relationer mellem de data, der kan karakteriseres som vigtige data, skal være repræsenteret i modellen.. ER-modelleringsteknik er vigtig for enhver database og danner grundlag for denne metode.

### E/R model<sup>4</sup>

En ER model er en abstrakt måde at beskrive en database på. Diagrammer skabt til at designe enheder og relationer kaldes entity-relationship diagrammer eller ER diagrammer.

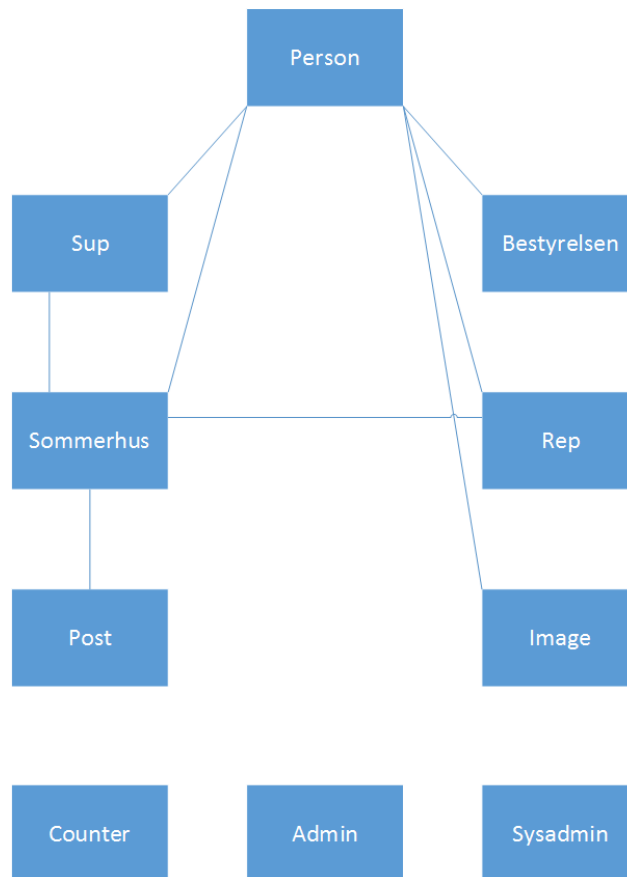
Ved selve designfasen af datasen kan der arbejdes med 3 niveauer af ER modeller.

Niveau 1: Conceptual data model

Denne model indeholder mindst mulig information om databasen, men samtidig viser den, hvilke elementer modellen skal indeholde såsom entiteter og relationships. Denne model kan benyttes som udgangspunkt for niveau 2, som er en logisk data model.

---

<sup>4</sup> Connolly (2010), kapitel 12



Figur 4 - Conceptual Data Model

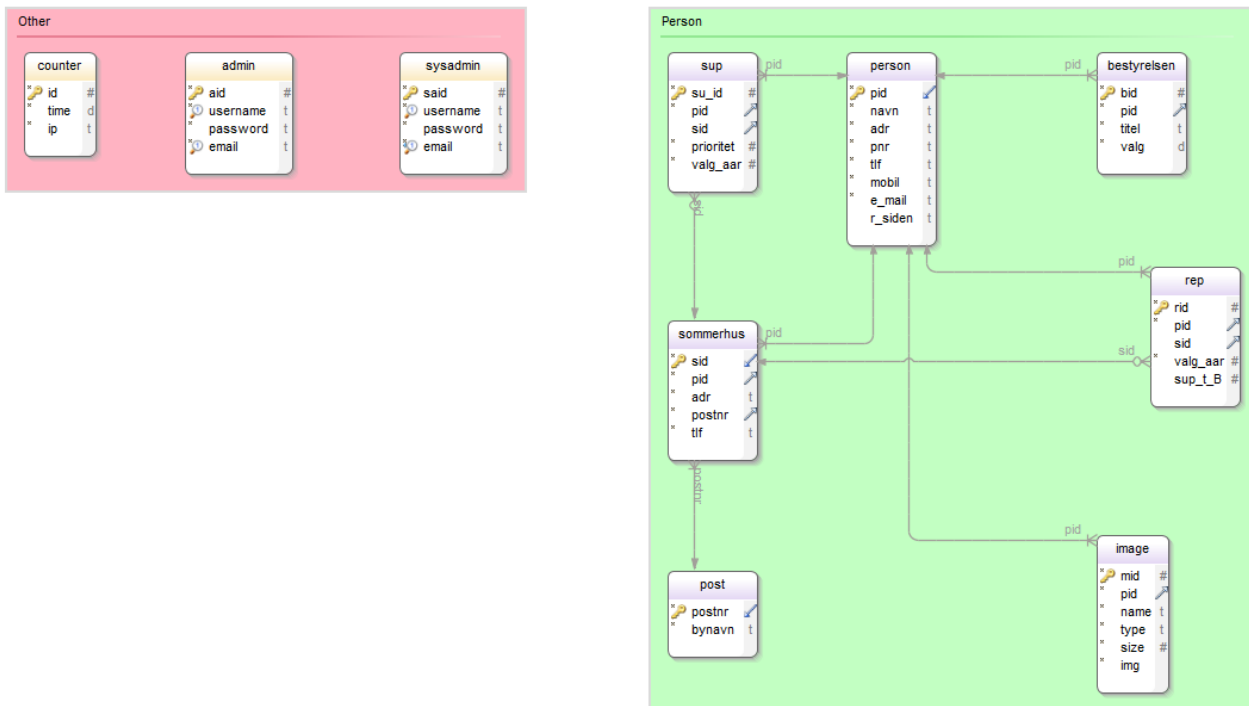
## Niveau 2: Logical data model

Logical data model er mere detaljeret i forhold til conceptual data modellen. Her er der tilføjet operationelle enheder og transaktionsdata. Detaljerne i de enkelte dataenheder udvikles, og enhedernes indbyrdes relationer bliver etableret. Dette indebærer tabeller, attributter, relationships og nøgler.

Symbol	Betydning
	Primær nøgle
	Fremmed nøgle
	Unik indeks
	Refererer til
	Refereret af
*	Not null
#	Int
T	Varchar
D	Date

Tabel 1 - Database Symbolbetydning

Databasescriptet<sup>5</sup> illustrerer integritetsbegrænsninger, som er nødvendige for databasen. Disse begrænsninger er specificeret som en del af databasens skema. I nedenstående figur 5 kan det ses, at der er specificeret både domænebegrænsninger og nøglebegrænsninger. Disse begrænsninger angiver de betingelser, der skal være opfyldt for enhver instans af databasen.



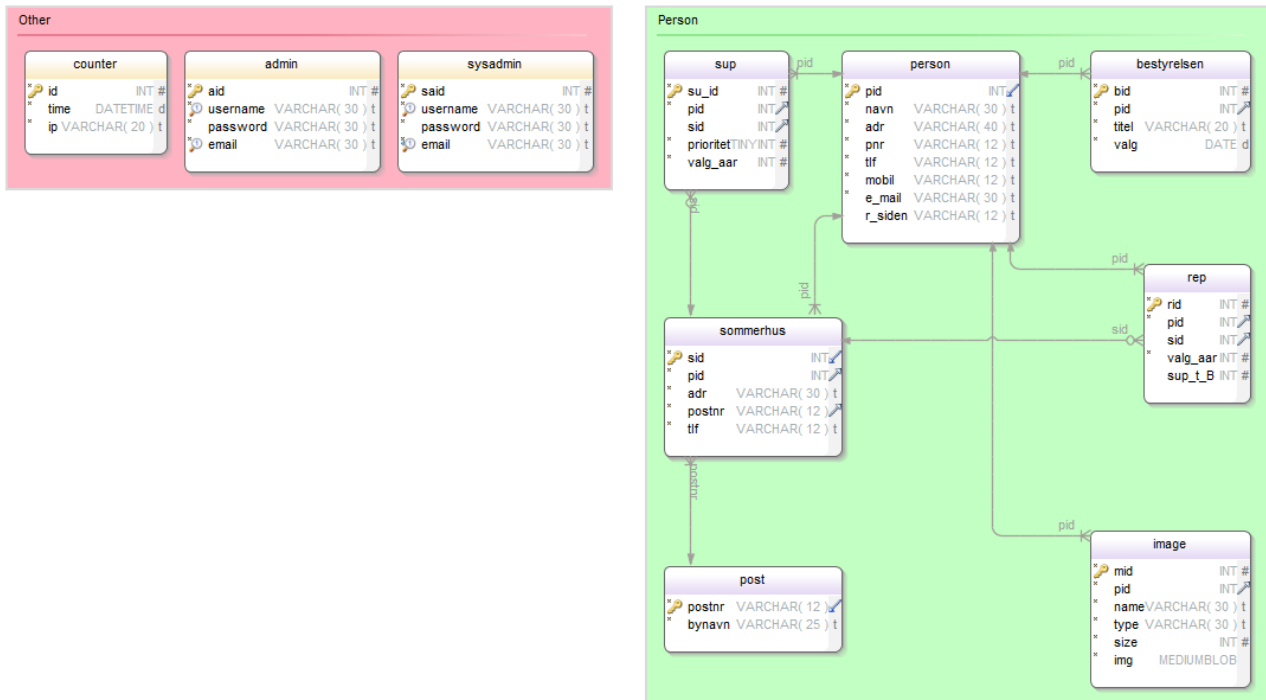
Figur 5 - Logical Data Model

<sup>5</sup> Databasescriptet kan findes i CD'en

### Niveau 3: Physical data model

Physical data model benyttes til at kunne instantiere databasen. Derfor er kravet til denne model, at modellen indeholder alle nødvendige detaljer, således der kan installeres en database. Dette indebærer tabeller, søjler, nøgler/indeks og datatyper.

Ud fra nedenstående figur kan databasen instantieres og benyttes. Modellen indeholder alle nødvendige tabeller, attributter, primære nøgler, fremmed nøgler, datatyper, indekser og relationer.



Figur 6 - Physical Data Model

### Normalisering<sup>6</sup>

Normalisering bruger en række tests (normal form) for at hjælpe med at identificere den optimale gruppe for disse attributter, for i sidste ende at identificere et egnet sæt relationer, der understøtter datakravene for virksomheden. Formålet med normalisering er derfor:

- At finde det minimale antal af attributter, som er nødvendige for at opfylde datakravene for virksomheden.
- At finde attributter i det samme forhold med et tæt logisk forhold.
- Minimal redundans, hvor hver attribut kun repræsenteres en gang med undtagelse af de attributter, der udgør hele eller en del af vigtige fremmede nøgler, som er afgørende for sammenføje af relaterede forbindelser.

<sup>6</sup> Connolly (2010), kapitel 14

### 1. Normalform

- Ikke repeterende attributter
- En primærnøgle i hver entitet, som entydigt udpeger en række

### 2. Normalform

- 1. Normalform
- Attributterne skal være afhængige af hele primærnøglen og ikke en del af primærnøglen

### 3. Normalform

- 2. Normalform
- Ingen transitive afhængigheder

Den implementerede database opfylder 3. normalform, da reglerne for 1. normalform, 2. normalform samt 3. normalform er gældende. Admin og sysadmin tabellen opfylder 2. normalform, da der er transitive afhængigheder mellem primær nøglen (id), brugernavn og e-mail. På grund af projektets størrelse og overskuelighed, så er det besluttet at holde alt admin og sysadmin informationer i en tabel for sig selv. For at implementere admin og sysadmin i 3. normalform, så må der ikke være transitive afhængigheder, og dette kunne implementeres ved at fjerne id feltet (primærnøglen) og gøre brugernavnet til primær nøglen. Dette vil også sikre at 2 brugere med fælles e-mail adresse kan oprettes i systemet.

I 1. normalform er der ingen repeterende attributter, men i hver entitet er der en primærnøgle, som entydigt udpeger en række. I 2. normalform er attributterne afhængige af hele primærnøglen. I 3. normalform er der ingen transitive afhængigheder.

# 9. Implementering

## 9.1. Use cases

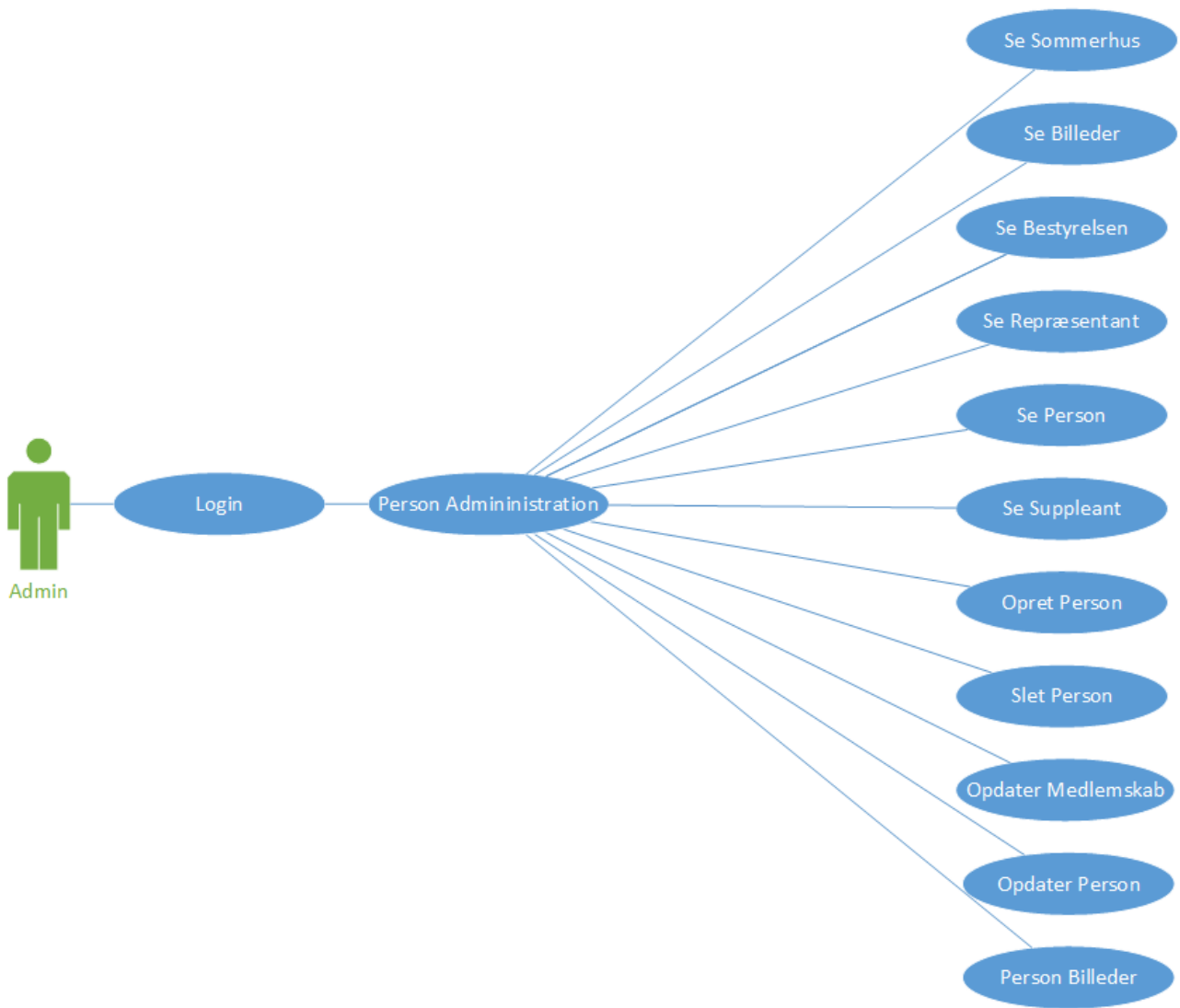
I dette afsnit beskrives de forskellige use cases som er gældende for systemet. De forskellige use cases er inddelt i en overordnet use case samt et par andre mere detaljerede use cases. Der er lavet use cases baseret på de to forskellige aktører, hvilket er admin og system admin.

### Main Use Case (admin)

Denne use case beskriver, hvilke overordnede muligheder admin aktøren har i systemet.

Aktøren skal først logge ind med gyldigt brugernavn og password. Derefter valideres det indtastede brugernavn samt password, og hvis de indtastede data er gyldige, så viderestilles aktøren til Person administrati-  
onssiden. På denne side har aktøren en række valgmuligheder.

1. Se sommerhus – her kan aktøren se alle informationer om personers sommerhuse.
2. Se billeder – her kan aktøren se informationer om alle personer og uploadede billeder af dem.
3. Se bestyrelsen – her kan aktøren se informationer om alle, som sidder i bestyrelsen.
4. Se repræsentant – her kan aktøren se informationer om alle personer, som sidder i repræsentantskabet.
5. Se person – her kan aktøren se informationer om samtlige personer.
6. Se suppleant – her kan aktøren se informationer om alle personer, som sidder i suppleantskabet.
7. Opret person – her kan aktøren oprette en ny person i systemet ved at indtaste de nødvendige data om personen.
8. Slet person – her kan aktøren slette personen helt fra systemet, således at personen ikke længere findes i systemet.
9. Opdater medlemskab – her kan aktøren tilføje/fjerne/erstatte personer i de ledende organer (suppleant, repræsentant, bestyrelse).
10. Opdater person – her kan aktøren opdatere/rette data for hver enkelt person.
11. Personbilleder – her kan aktøren tilføje/fjerne billeder af den pågældende person.



Figur 7 - Use Case - Main Use Case - Admin



## Main Use Case (Sys admin)

Denne use case beskriver, hvilke overordnede muligheder sys admin-aktøren har i systemet.

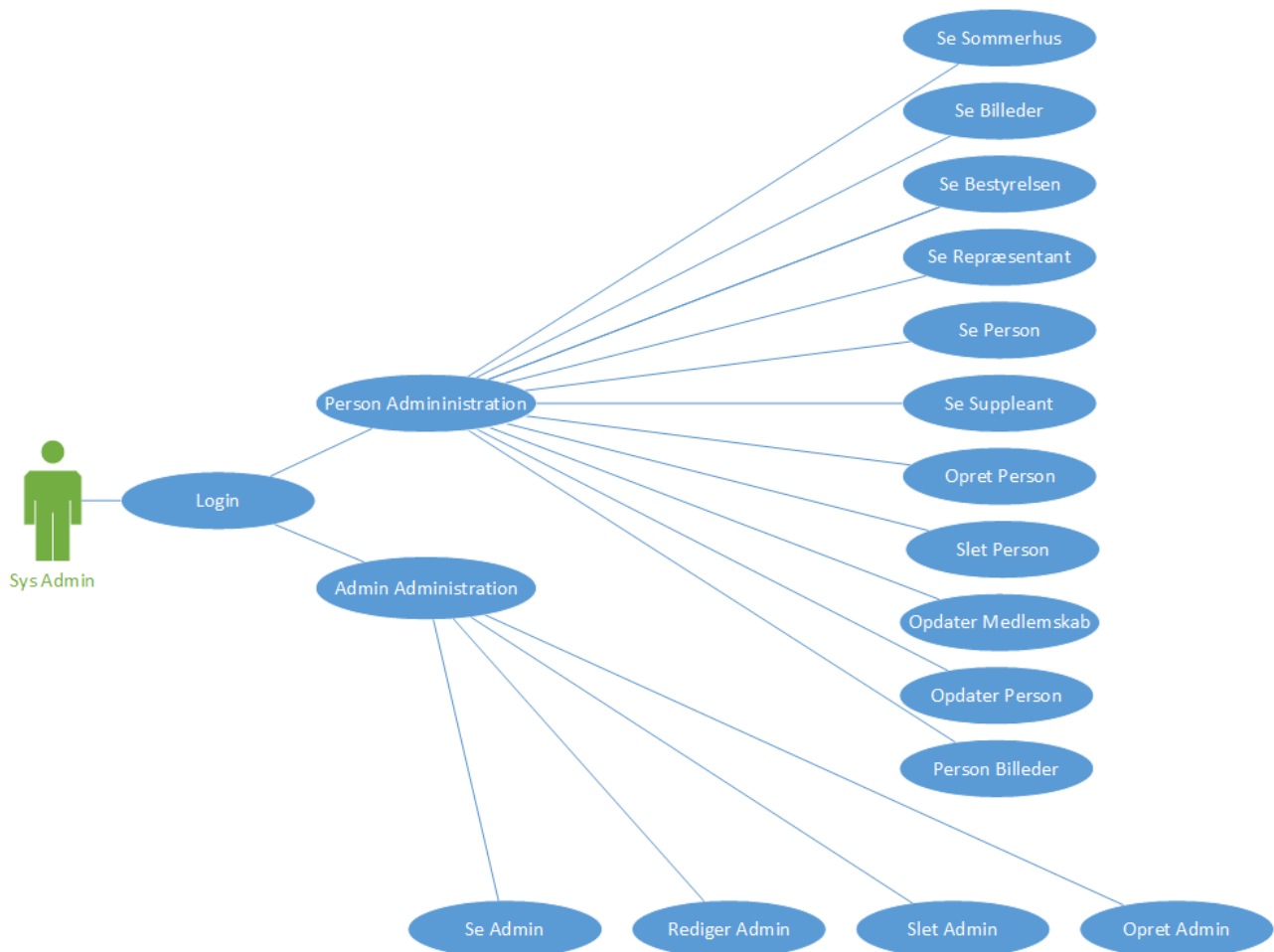
Aktøren skal først logge ind med gyldigt brugernavn og password. Derefter valideres det indtastede brugernavn samt password, og hvis de indtastede data er gyldige, har aktøren adgang til person-administrations-siden og admin-administrationssiden.

### Person administration

1. Se sommerhus – her kan aktøren se alle informationer om personers sommerhuse.
2. Se billeder – her kan aktøren se informationer om alle personer samt uploadede billeder af dem.
3. Se bestyrelsen – her kan aktøren se informationer om alle personer, som sidder i bestyrelsen.
4. Se repræsentant – her kan aktøren se informationer om alle personer, som sidder i repræsentantskabet.
5. Se person – her kan aktøren se informationer om alle personer.
6. Se suppleant – her kan aktøren se informationer om alle personer, som sidder i suppleantskabet.
7. Opret person – her kan aktøren oprette en ny person i systemet ved at indtaste de nødvendige data om personen.
8. Slet person – her kan aktøren slette personen helt fra systemet, således at personen ikke længere findes i systemet.
9. Opdater medlemskab – her kan aktøren tilføje/fjerne/erstatte personer i de ledende organer (suppleant, repræsentant, bestyrelsen).
10. Opdater person – her kan aktøren opdatere/rette data for hver enkelt person.
11. Person billeder – her kan aktøren tilføje/fjerne billeder for den pågældende person.

### Admin administration

1. Se admin – her kan aktøren se informationer om alle admin-brugere.
2. Rediger admin – her kan aktøren redigere i admin-brugerens informationer.
3. Slet admin – her kan aktøren slette admin-brugere fra systemet.
4. Opret admin – her kan aktøren tilføje nye admin-brugere til systemet.



Figur 8 - Main Use Case - Sysadmin

### Use case – Opret person (admin)

Denne use case omhandler admin-aktørens muligheder, hvad angår oprettelse af personer i systemet. Aktøren kan således oprette 3 forskellige brugere:

1. Opret suppleant – her kan aktøren oprette en suppleant i systemet ved at udfylde de nødvendige felter med persondata.
2. Opret alm. person – her kan aktøren oprette en almindelig person, som ikke er tilknyttet de ledende organer.
3. Opret repræsentant – Her kan aktøren oprette en repræsentant i systemet ved at udfylde de nødvendige felter med persondata.

Use case diagrammet kan ses i bilagsafsnittet Figur 46

### **Use case – Slet person (admin)**

Denne use case omhandler admin-aktørens muligheder, hvad angår slettelse af personer i systemet. Aktøren har kun en mulighed:

1. Slet person – her kan aktøren slette en person helt fra systemet, så personen ikke længere findes i systemet.

Use case diagrammet kan ses i bilagsafsnittet Figur 47

### **Use case – Opdater medlemskab (admin)**

Denne use case omhandler admin-aktørens muligheder, hvad angår at tilføje/fjerne/erstatte personer, hvad enten de sidder i de ledende organer eller ej. Dette er muligt for alle 3 poster:

1. Suppleant – her kan aktøren tilføje en person til suppleantposten, fjerne en person fra suppleantposten.
2. Repræsentant – her kan aktøren tilføje en person til repræsentantposten, fjerne en person fra repræsentantposten.
3. Bestyrelsen – her kan aktøren tilføje en person til bestyrelsesposten, fjerne en person fra bestyrelsesposten.
4. Find – hvis personen sidder i repræsentantskabet eller bestyrelsen, er det muligt at erstatte personen med en egnet suppleant.

Use case diagrammet kan ses i bilagsafsnittet Figur 48

### **Use case – Opdater person (admin)**

Denne use case omhandler admin-aktørens muligheder, hvad angår redigering/opdatering af data, som er gældende. Det vil kun være muligt at redigere i data, som personen er medlem i.

1. Person data – Her kan aktøren redigere/opdatere den pågældende persons grunddata.
2. Sommerhus data – Her kan aktøren redigere/opdatere den pågældende persons sommerhusdata.
3. Repræsentant data – Her kan aktøren redigere/opdatere den pågældende persons repræsentantdata.
4. Suppleant data – Her kan aktøren redigere/opdatere den pågældende persons suppleantdata.
5. Bestyrelse data – Her kan aktøren redigere/opdatere den pågældende persons bestyrelsesdata.

Use case diagrammet kan ses i bilag afsnittet Figur 49

### **Use case – Person billeder (admin)**

Denne use case omhandler admin aktørens muligheder hvad angår tilføjelse/fjernelse af billeder for den pågældende person. Hver person har ret til at få tilføjet maksimalt 1 billede.

1. Fjern billede – her kan aktøren fjerne den pågældende persons billede fra systemet.
2. Tilføj billede – her kan aktøren tilføje et billede af den pågældende person til systemet.

Use case diagrammet kan ses i bilagsafsnittet Figur 50

### **Use case – Se (admin)**

Denne use case er beskrevet i main use case (admin), og er derfor ikke medtaget i dette afsnit.

### **Use case – Opret admin (sys admin)**

Denne use case er beskrevet i main use case (sysadmin), og er derfor ikke medtaget i dette afsnit.

### **Use case – Slet admin (sys admin)**

Denne use case er beskrevet i main use case (sysadmin), og er derfor ikke medtaget i dette afsnit.

### **Use case – Rediger admin (sys admin)**

Denne use case er beskrevet i main use case (sysadmin), og er derfor ikke medtaget i dette afsnit.

### **Use case – Se admin (sys admin)**

Denne use case er beskrevet i main use case (sysadmin), og er derfor ikke medtaget i dette afsnit.

## 9.2. Flowchart

Flowchart-diagrammer giver en mere detaljeret beskrivelse af, hvilke processer der foregår i systemet. Dette sikrer en visualisering af processerne samt en bedre viden om og forståelse af, hvordan systemet fungerer fra start til slut.

### Flowchart – Opret person (admin)

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når en ny person ønskes oprettet i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login-informationer valideres. Hvis informationerne er valide, viderestilles brugeren til brugeradministrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Opret person – brugeren vælger at oprette en ny person.
4. Person/suppleant/repræsentant – Brugeren bedes afgøre, hvilken type person der ønskes oprettet. Her er valget mellem en almindelig person, en suppleant og en repræsentant. Baseret på valget ændrer systemet inputfelterne, så de relevante inputfelter vises.
5. Indtast data – brugeren indtaster data for den nye person, der skal oprettes.
6. Submit/input valid – de indtastede data bliver først validerede. Derefter bliver de indsendt til serversiden, dvs. de bagvedkørende scripts.
7. Check login – på serversiden kontrolleres atter, om login-dataene er valide. Hvis ikke, viderestilles man til login-siden. Er de valide, viderestilles man til Adduser-siden.
8. Adduser – her kontrolleres igen, hvilken type bruger der ønskes oprettet. Valget står mellem en almindelig person, en suppleant og en repræsentant. Personen oprettes også i sommerhustabellen enten som fastboende eller ikke-fastboende, baseret på, om sommerhus-felterne er tomme eller udfyldte.
9. Rorvigvand DB – i Adduser klargøres de forskellige funktioner, som derefter skriver ned i databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 51.

## Flowchart – Slet person (admin)

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når en person ønskes slettet i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login-informationer valideres. Hvis informationerne er valide, viderestilles brugeren til brugeradministrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Slet person – brugeren vælger at slette en person fra systemet.
4. Valg af person – brugeren vælger, hvilken person der skal slettes fra systemet gennem en drop down-menu.
5. Submit – dataene bliver indsendt til serversiden, dvs. de bagvedkørende scripts.
6. Check login – på serversiden kontrolleres på ny, om login-dataene er valide. Hvis ikke, viderestilles man til login-siden. Er de valide, viderestilles man til Deluser-siden.
7. Deluser – her kontrolleres igen, hvilken type bruger der ønskes slettet. Valget står mellem en almindelig person, en suppleant og en repræsentant. Deluser sletter også sommerhus-data og billede-data. Deluser sletter fra brugeren fra alle tabeller.
8. Rorvigvand DB – i Deluser klargøres de forskellige funktioner, som derefter sletter dataene fra databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 52.

## Flowchart – Opdater medlemskab (admin)

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når medlemskabet for en person ønskes opdateret i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login-informationer valideres. Hvis informationerne er valide, viderestilles brugeren til brugeradministrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Opdater medlemskab – brugeren vælger at opdatere medlemskabet for en person i systemet.
4. Valg af person – brugeren vælger, hvilken person der skal opdateres i systemet gennem en drop down-menu.
5. Bestyrelse/suppleant/repræsentant – brugeren bliver bedt om at afgøre, hvilken type person der ønskes opdateret. Her står valget mellem en almindelig person, en suppleant og en repræsentant. Systemet vil her afgøre om personen er oprettet i de pågældende tabeller (bestyrelse og repræsentant).
6. Allerede oprettet – hvis brugeren allerede er oprettet i tabellerne, vises en fjern og find-knap for den pågældende gruppe. Hvis ikke, vises de nødvendige input-felter.
7. Indtast data – brugeren indtaster data for den pågældende person, der skal opdateres.
8. Submit/input valid – de indtastede data bliver først validerede. Derefter bliver de indsendt til serversiden, dvs. de bagvedkørende scripts.
9. Check login – på serversiden kontrolleres på ny, om login-dataene er valide. Hvis ikke, viderestilles man til login-siden. Er de korrekte, viderestilles man til en af Addbes/Addsup/Addrep-siderne.
10. Addbes/Addsup/Addrep – her køres de nødvendige funktioner for den pågældende person som tilføjer dataene til personen.
11. Rorvigvand DB – på en af Addbes/Addsup/Addrep-siderne klargøres de forskellige funktioner, som derefter skriver ned i databasen.
12. Fjern/Find – brugeren kan vælge mellem fjern eller find. Ved valg af fjern viderestilles man til en af delsup/delbes/delrep-siderne. Ved valg af find viderestilles man til find-siden.
13. delsup/delbes/delrep – her køres de nødvendige funktioner for den pågældende person, som sletter dataene for personen. Delsup sletter personen fra suppleant tabellen, delbes sletter personen fra bestyrelsestabellen og delrep sletter personen fra repræsentanttabellen.
14. Find– her køres de nødvendige funktioner for den pågældende person, som finder den bedst mulige afløser (suppleant) til personen. Der bliver prioriteret efter samme kreds og samme valgår og ellers en tilfældig afløser (suppleant).
15. Rorvigvand DB – funktionerne eksekveres i databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 53.

## Flowchart – Opdater person (admin)

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når persondata ønskes opdateret i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login-informationer valideres. Hvis informationerne er valide, viderestilles brugeren til brugeradministrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Opdater person – brugeren vælger at opdatere dataene for en person i systemet.
4. Valg af person – brugeren vælger, hvilken person der skal opdateres i systemet gennem en drop down-menu.
5. Bestyrelse/suppleant/repræsentant/person/sommerhus – Systemet viser kun redigerbare felter for de grupper, man er medlem af.
6. Rediger data – bruger kan hermed redigere/opdatere dataene i de pågældende input-felter.
7. Submit/input valid – de indtastede data bliver først validerede. Derefter bliver dataene indsendt til serversiden, dvs. de bagvedkørende scripts.
8. Check login – på serversiden kontrolleres på ny, om login-dataene er valide. Hvis ikke, viderestilles man til login-siden. Er de valide, viderestilles man til ediusersiden.
9. Ediusersiden – her køres de nødvendige funktioner, som opdaterer dataene for de grupper, som personen er medlem af.
10. Bestyrelse/suppleant/repræsentant/person/sommerhus – Opdateringer udfærdiges for alle de grupper, som personen er medlem af.
11. Rorvigvand DB – Dataene opdateres i databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 54.



## Flowchart – Person billeder (admin)

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når et billede af en person ønskes tilføjet/slettet/set i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login-informationer valideres. Hvis informationer er valide, viderestilles brugeren til brugeradministrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Person billeder – brugeren vælger at tilføje/slette/se billedet af en person i systemet.
4. Valg af person – brugeren vælger, hvilken person der skal opdateres i systemet gennem en drop down-menu.
5. Findes billede – systemet afgør, om der allerede findes et billede af den pågældende person. I bekræftende fald viderestilles man til delimg-siden, hvor billedet samtidig bliver vist. I benægtende fald viderestilles man til addimg-siden, hvor der er mulighed for at uploade et billede.
6. Upload – brugeren kan hermed uploade et billede for den pågældende person.
7. Submit – dataene bliver indsendt til serversiden, dvs. de bagvedkørende scripts (addimg).
8. Check login – på serversiden kontrolleres på ny, om login-dataene er valide. Login-dataene kontrolleres for begge scenarier (findes billede).
9. Addimg – her køres de nødvendige funktioner, som tilføjer et billede af personen.
10. Delimg eller se billede – brugeren kan se det eksisterende billede samt fjerne det. I Delimg køres de nødvendige funktioner, som sletter billedet af personen.
11. Rorvigvand DB – Dataene opdateres i databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 55.

## Flowchart – Opret admin (sysadmin)

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når en ny admin-bruger ønskes oprettet i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login informationer valideres. Hvis informationer er valide, viderestilles brugeren til admin-administrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Opret admin – brugeren vælger at oprette en ny admin-bruger.
4. Indtast data – brugeren indtaster data for den nye admin-bruger, der skal oprettes.
5. Submit/input valid – de indtastede data bliver først validerede. Derefter bliver dataene indsendt til serversiden, dvs. de bagvedkørende scripts.
6. Check login – på server siden kontrolleres på ny, om login-dataene er valide. Hvis ikke, viderestilles man til login-siden. Er de valide, viderestilles man til Addadmin-siden.
7. Addadmin – de udfyldte felter hentes og indsættes i mySQL-forespørgsler.
8. Rorvigvand DB – i Addadmin eksekveres de forskellige funktioner, som skriver til databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 56.

### **Flowchart – Slet admin (sysadmin)**

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når en admin-bruger ønskes slettet i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login-informationer valideres. Hvis informationer er valide, viderestilles brugeren til admin-administrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Slet admin – brugeren vælger at slette en admin-bruger fra systemet.
4. Valg af admin bruger – brugeren vælger, hvilken admin-bruger der skal slettes fra systemet gennem en drop down-menu.
5. Submit – dataene bliver indsendt til serversiden, dvs. de bagvedkørende scripts.
6. Check login – På server siden kontrolleres på ny, om login-dataene er valide. Hvis ikke, viderestilles man til login siden. Er de valide, viderestilles man til Deladmin-siden.
7. Deladmin – her køres en række tjek, og derefter klargøres en MySQL-forespørgsel.
8. Rorvigvand DB – i Deladmin eksekveres de forskellige funktioner, som derefter sletter dataene fra databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 57.

### **Flowchart – Rediger admin (sysadmin)**

Dette flowchart visualiserer, hvilke processer brugeren gennemgår fra start til slut, når redigering af en admin-bruger ønskes udført i systemet.

1. Login – brugeren indtaster login-informationer på login-siden.
2. Login valid – de indtastede login-informationer valideres. Hvis informationer er valide, viderestilles brugeren til admin-administrationsmodulet, ellers bliver brugeren bedt om at indtaste informationerne igen.
3. Rediger admin – brugeren vælger at opdatere dataene for en admin-bruger i systemet.
4. Valg af admin brugeren – brugeren vælger, hvilken admin-bruger der skal have opdateret sine data.
5. Rediger data – bruger kan hermed redigere/opdatere dataene i de pågældende inputfelter.
6. Submit/input valid – De indtastede data bliver først validerede. Derefter bliver dataene indsendt til serversiden, dvs. de bagvedkørende scripts.
7. Check login – På serversiden kontrolleres på ny, om login-dataene er valide. Hvis ikke, viderestilles man til login side. Er de valide, viderestilles man til ediaadmin-siden.
8. Ediaadmin – De udfyldte felter hentes og sættes ind i MySQL forespørgsler.
9. Rorvigvand DB – Dataene opdateres i databasen.

Flowchart diagrammet kan ses i bilagsafsnittet Figur 58.

### 9.3. Sekvensdiagram

I dette afsnit beskrives de forskellige sekvensdiagrammer som er gældende for systemet. De forskellige sekvensdiagrammer er inddelt på baggrund på hver use case.

Sekvensdiagrammer er interaktions diagrammer som viser hvordan processer opererer med andre processer samt i hvilken rækkefølge.

#### **Sekvensdiagram – Opret person (admin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når en ny person ønskes oprettet i systemet. Det er forudsat, at brugeren er logget ind med administratorrettigheder.

1. Klient indtaster persondata, som valideres og sendes til adduser-siden.
2. Adduser-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
3. Adduser siden skriver de nye data i databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 59.

#### **Sekvensdiagram - Slet person (admin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når en person ønskes slettet i systemet. Det er forudsat, at brugeren er logget ind med administratorrettigheder.

1. Klient kalder databasen og ønsker hele personlisten.
2. Databasen sender personlisten til klienten.
3. Klienten vælger, hvilken person der skal slettes fra listen, som sendes til deluser-siden.
4. Deluser-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
5. Deluser siden sletter dataene fra databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 60.

### **Sekvensdiagram – Opdater medlemskab (admin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når medlemskab for en person ønskes opdateret i systemet. Det er forudsat, at brugeren er logget ind med administratorrettigheder.

1. Klienten kalder databasen og ønsker hele personlisten.
2. Databasen sender personlisten til klienten.
3. Klienten vælger, hvilken person der skal opdateres fra listen, som sendes til getmember-siden.
4. Getmember-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
5. Getmember-siden returnerer medlemskabsdata for den pågældende person og returnerer de relevante felter.
6. Klienten udfører en aktion for et givet felt, og dataene valideres først og bliver derefter sendt til enten add, fjern eller find, baseret på, hvilken aktion klienten har udført.
7. Del/add/find-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
8. Del/add/find-siden skriver/sletter dataene i databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 61.

### **Sekvensdiagram – Opdater person (admin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når persondata for en person ønskes opdateret i systemet. Det er forudsat, at brugeren er logget ind med administratorrettigheder.

1. Klienten kalder databasen og ønsker hele personlisten.
2. Databasen sender personlisten til klienten.
3. Klienten vælger, hvilken person der skal opdateres fra listen, som sendes til getuser-siden.
4. Getuser-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
5. Getuser-siden returnerer persondata for den pågældende person og returnerer de relevante felter.
6. Klienten udfører en aktion ved at redigere i de forskellige felter, hvorefter dataene først valideres og derefter sendes til ediuser-siden.
7. Ediuser-siden kontrollerer, at login sessionen er gyldig med login-siden.
8. Ediuser siden skriver/redigerer dataene i databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 62.

### **Sekvensdiagram – Personbilleder (admin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når der ønskes at tilføje/slette/se billede af en person i systemet. Det er forudsat, at brugeren er logget ind med administratorrettigheder.

1. Klienten kalder databasen og ønsker hele personlisten.
2. Databasen sender personlisten til klienten.
3. Klienten vælger, hvilken person der skal opdateres fra listen, som sendes til getimg-siden.
4. Getimg-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
5. Getimg-siden henter billeddata for den pågældende bruger fra databasen.
6. Getimg-siden returnerer billeddata for den pågældende person og returnerer de relevante felter.
7. Klienten udfører en aktion ved enten at slette et billede eller uploade et nyt baseret på, om der allerede eksisterer et billede eller ej i databasen. Derefter sendes dataene enten til delimg eller addimg.
8. Delimg/addimg-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
9. Delimg sletter billeddataene for personen i databasen og addimg tilføjer billeddataene for personen i databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 63.

### **Sekvensdiagram – Login (sysadmin og admin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når en klient forsøger at logge ind i systemet som admin-bruger eller sysadmin-bruger.

1. Klient trykker på "Login" i menuen og bliver viderestillet til login-siden.
2. Klient indtaster brugernavn og kodeord.
3. Login siden kalder Authenticate-siden.
4. Authenticate henter brugernavn og kodeord. Det kontrolleres, om brugernavn/kodeord-parret findes i databasen.
5. Authenticate-siden sætter sessionvariablen og kalder enten authenticate\_succes eller authenticate\_sysadmin\_succes baseret på om der er logget ind med en admin-bruger eller en sysadmin-bruger.
6. Authenticate\_succes/authenticate\_sysadmin\_succes sætter sessionvariablen til at være sand.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 64.

### **Sekvensdiagram – Glemt kodeord (admin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når en admin-bruger har glemt sit kodeord til systemet.

1. Klient trykker på "glemt kodeord"-linket og bliver viderestillet til e-mail siden.
2. På e-mail siden indtaster klienten sit brugernavn.
3. Brugernavnet sendes til server-siden recover.
4. Recover-siden tager imod brugernavnet og henter relevante data fra databasen baseret på brugernavnet.
5. Databasen returnerer dataene til recover-siden.
6. Recover-siden kalder mailfunktionen og udfylder parametrene med de relevante data og kalder SMTP-serveren.
7. SMTP-serveren sender en e-mail med brugernavn og kodeord til brugerens e-mailadresse.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 65.

### **Sekvensdiagram – Opret admin (sysadmin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når en ny admin-bruger ønskes oprettet i systemet. Det er forudsat, at brugeren er logget ind med sysadmin-rettigheder.

1. Klient indtaster persondata, som valideres og sendes til addadmin-siden.
2. Addadmin-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
3. Addadmin-siden skriver de nye data i databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 66.

### **Sekvensdiagram - Slet admin (sysadmin)**

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når en admin bruger ønskes slettet i systemet. Det er forudsat, at brugeren er logget ind med sysadmin-rettigheder.

1. Klient kalder databasen og ønsker hele admin-listen.
2. Databasen sender admin-listen til klienten.
3. Klienten vælger, hvilken admin-bruger der skal slettes fra listen, som sendes til deladmin-siden.
4. Deladmin-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
5. Deladmin-siden sletter dataene fra databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 67.

## Sekvensdiagram – Opdater admin (sysadmin)

Dette sekvensdiagram visualiserer interaktionen mellem processerne, når data for en admin-bruger ønskes opdateret i systemet. Det er forudsat at brugeren er logget ind med sysadmin-rettigheder.

1. Klienten kalder databasen og ønsker hele admin-listen.
2. Databasen sender admin-listen til klienten.
3. Klienten vælger, hvilken admin-bruger der skal opdateres fra listen, som sendes til getadmin-siden.
4. Getadmin-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
5. Getadmin-siden returnerer admin-data for den pågældende admin-bruger og returnerer de relevante felter.
6. Klienten udfører en aktion ved at redigere i de forskellige felter, hvorefter dataene først valideres og derefter sendes til ediadmin-siden.
7. Ediadmin-siden kontrollerer, at login-sessionen er gyldig i forhold til login-siden.
8. Ediadmin-siden skriver/redigerer dataene i databasen.

Sekvensdiagrammet kan ses i bilagsafsnittet Figur 68.

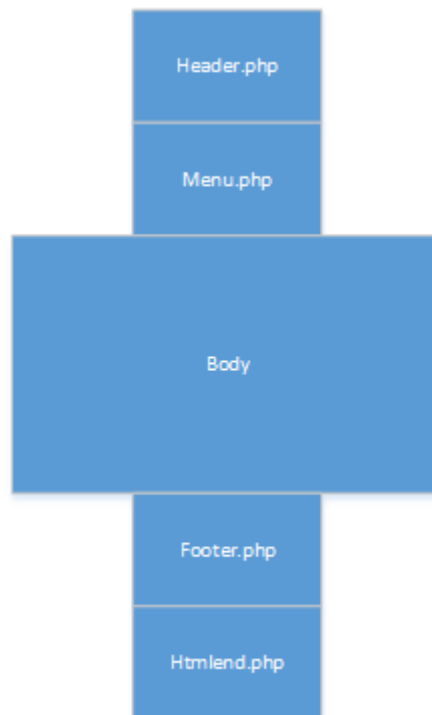
## 9.4. Frontend/Backend

I webudvikling tales der om frontend-udvikling og backend-udvikling. Frontend er en abstraktion, som muliggør, at underliggende komponenter fremstilles til brugeren på en brugervenlig måde. Dette omfatter HTML, CSS og javascript, hvor selve brugergrænsefladen defineres. Frontend præsenterer indholdet på en brugervenlig måde, så det fremstår pænt for brugeren.

Backend derimod udgøres af de underliggende komponenter, som ligger på serveren, hvor selve funktionerne og beregningerne optræder, såsom brugerautorisation, login-funktionalitet og funktioner til fremvisning af informationer på frontend-siden. Når en bruger ønsker en handling udført, sendes ønsket gennem et sikkerhedstjek, hvorefter backend-funktioner køres, og en webside bliver genereret og sendt retur til brugeren via en webserver. Backend-delen udgøres af PHP scripts og diverse funktioner, herunder funktioner, som henter/sender informationer til databasen.

## Frontend

Den overordnede struktur på denne webside består af en header, en menu, body (indhold for den pågældende side), footer samt htmlend.



Figur 9 - Web Struktur

### header.php

Denne fil indeholder HTML, som udgør header-delen på websiden. Header.php indgår på samtlige php-sider. I header-delen findes flere forskellige informationer, herunder:

Dokument type: bruges til at give browseren information om, i hvilken HTML-version den pågældende side forefindes i.

Metadata: bruges for at give browseren information om data

Link: bruges som reference til CSS-filen.

Link: bruges også til at give dokumentet et ikon.

Title: bruges til at give dokumentet en titel



## **Menu.php**

Denne fil indeholder HTML, som udgør menuen på websiden. Menuen er implementeret som en drop down-menu med undermenuer, der foldes ud ved at holde musen over de forskellige elementer. Menuen indgår på samtlige php-sider.

## **Footer.php**

Denne fil indeholder en div-tag med copyright information. Footer indgår på samtlige php-sider.

## **htmlend.php**

Denne fil indeholder HTML med afslutning på body-tag og afslutning på html-tag, som udgør slutningen på HTML-koden og indgår på samtlige php-sider.

## **Historie.php, Index.php, Kontaktpersoner.php, Meddelelser.php, Webmaster.php, repræsentantskab.php og mødeplan.php**

Disse filer er bygget op efter webstrukturen på figur 9, hvor body'en indeholder tekst, billeder og links.

## **Medlemmer.php, suppleanter.php og bestyrelsen.php**

Disse filer er ligeledes bygget op efter webstrukturen på figur 9, hvor body'en indeholder tekst, billeder, links og data fra databasen.

## **Leftsidebar.php**

Også denne fil er bygget op efter webstrukturen på figur 9, hvor body'en indeholder data fra databasen om websidens besøgstal. Denne php-fil tæller antallet af besøgene på websiden, hvilket kan ses i webmaster.php

## Javascript – user\_management og admin\_management

Javascript funktion	Beskrivelse
Document.getElementById("div id").innerHTML	Indsætter response-tekst I det pågældende <div>
XMLHttpRequest()	API til at sende og modtage HTTP/HTTPS forespørgsler til/fra en webserver, som understøttes af IE7+, Chrome, Firefox, Opera og Safari
ActiveXObject	API til at sende og modtage http/HTTPa forespørgsler til/fra en webserver, som understøttes af IE5 og IE6
Onreadystatechange	Funktion som fungerer som en event-trigger, lytter på readyState og status
readyState	Holder på status fra XMLHttpRequest
Status	Statusbesked
Xmlhttp.responsetext	Responsebesked fra webserveren
Xmlhttp.open	Specificerer, hvilken type forespørgsel der skal sendes til webserveren
Xmlhttp.send	Sender forespørgslen

Tabel 2- Javascript funktioner

Der er implementeret javascript i user\_management.php filen til at varetage forskellige funktioner, som kun køres på klientsiden, input-validering på klientsiden, funktioner, som benyttes i forbindelse med at sende data til serversiden, samt funktioner, som benyttes i forbindelse med indhentning af data fra serversiden.

### Funktion showMember(str)<sup>7</sup>

Denne funktion benyttes i forbindelse med feltet opdater medlemskab. I opdater medlemskab-feltet skal man vælge en person fra select-listen, hvorefter javascript-metoden showMember(pid) kaldes og tager person-id som input-parameter. Metoden kaldes, hver gang select-listen ændres, og showMember-funktionen kaldes med det pågældende person-id. Der er ydermere tilføjet et <div id='member'> felt, som opdateres med data, hver gang showMember kaldes.

I showMember-funktionen køres der en række tjek på inputtet. Hvis input-strengen er tom, sættes <div id='member'> til at være tom. Såfremt klientens browser er IE7+, Firefox, Chrome, Opera eller Safari, instantieres der en XMLHttpRequest<sup>8</sup>. Hvis klientens browser er IE6 eller IE5 så instantieres et ActiveXObject<sup>9</sup>.

onreadystatechange instantieres og lytter på readyState og statusværdierne. Hver gang readyState er 4<sup>10</sup>, og status er 200<sup>11</sup>, indsættes responseteksten fra getmember.php filen i <div id='member'> feltet.

I xmlhttp.open skal det, hver gang en forespørgsel sendes til webserveren, specificeres, hvilken type forespørgsel der skal sendes til webserveren, og her vælges GET, getmember.php?q=pid.

<sup>7</sup> User\_management.php, linie: 13-36

<sup>8</sup> XMLHttpRequest er et API til at sende og modtage HTTP/HTTPS forespørgsler til/fra en webserver

<sup>9</sup> ActiveXObject tilbydes kun i ældre browsere for IE men tilbyder samme funktionalitet som XMLHttpRequest.

<sup>10</sup> 4=Forespørgsel gennemført og responset er klart

<sup>11</sup> 200=status er OK

Getmember.php filen er en server fil, som køres for at hente de data, som vedrører den valgte bruger.

### **Funktion showUser(str)<sup>12</sup>**

Denne funktion benyttes i forbindelse med feltet opdater bruger. I opdater medlemskab-feltet skal man vælge en person fra select-listen, hvorefter javascript-metoden showUser(pid) kaldes og tager person-id som input-parameter. Metoden kaldes, hver gang select-listen ændres, og showUser-funktionen kaldes med den pågældende person-id. Der er ydermere tilføjet et <div id='user'> felt, som opdateres med data, hver gang showUser kaldes.

Indholdet i showUser funktion er magen til showMember. Den eneste forskel findes i xmlhttp.open hvor forespørgslen til webserveren sættes til getuser.php?q=pid.

### **Funktion showImg(str)<sup>13</sup>**

Denne funktion benyttes i forbindelse med feltet personbilleder. I personbilleder-feltet skal man vælge en person fra select-listen, hvorefter javascript metoden showImg(pid) kaldes og tager person-id som input-parameter. Metoden kaldes for hver gang select-listen ændres og showImg-funktionen kaldes med den pågældende person-id. Der er ydermere tilføjet et <div id='img'> felt, som opdateres med data, hver gang showImg kaldes.

Indholdet i showImg funktion er magen til showMember. Den eneste forskel findes i xmlhttp.open, hvor forespørgslen til webserveren sættes til getimg.php?q=pid.

### **Funktion suppleantCheckbox(state)<sup>14</sup>**

Denne funktion benyttes i forbindelse med feltet opret person. I opret person kan man afkrydse feltet suppleant, og derved vil der fremstå to ekstra input-felter. Funktionen suppleantCheckbox(value) kaldes, hver gang brugeren klikker på checkboksen ud fra suppleantfeltet. Funktionen kaldes med parameteren value<sup>15</sup>. Der er ydermere tilføjet et <div id='supp\_valg'> felt og <div id='supp\_prio'> felt, som opdateres, hver gang suppleantCheckbox-funktionen kaldes.

Først tjekkes værdien på input-parameteren. Hvis denne er sand, sættes en global variabel supChecked til at være sand (denne variabel benyttes til input validering).

Hvis både supChecked og repChecked er sat til at være sande, kommer der en fejlmeddelelse fra systemet om, at der både ved repræsentantfeltet og ved suppleantfeltet er sat flueben. Derefter fjernes begge felters flueben, og de globale variabler supChecked og repChecked sættes til at være falske – hvorefter de to <div id='supp\_valg'>, <div id='supp\_prio'> og <div id='repp\_valg'>, <div id='repp\_sup'> sættes til at være tomme.

Hvis mindst en af variablerne supChecked eller repChecked er falske, opdateres <div id='supp\_valg'>, <div id='supp\_prio'> med to nye input felter.

---

<sup>12</sup> User\_management.php, linie: 40-64

<sup>13</sup> User\_management.php, linie: 68-92

<sup>14</sup> User\_management.php, linie: 98-122

<sup>15</sup> Value kan enten være sand eller falsk. Sand=checked, falsk=ikke checked.

Hvis værdien på input parameteren er falsk, opdateres <div id='supp\_valg'>, <div id='supp\_prio'> til at være tomme.

### **Funktion repræsentantCheckbox(state)<sup>16</sup>**

Denne funktion benyttes i forbindelse med feltet opret person. I opret person kan man afkrydse feltet repræsentant, og derved vil der fremstå to ekstra input-felter. Funktionen repræsentantCheckbox(value) kaldes, hver gang brugeren klikker på checkboksen ud fra repræsentantfeltet. Funktionen kaldes også med parameteren value. Der er ydermere tilføjet et <div id='repp\_valg'> felt og <div id='repp\_sup'> felt, som opdateres, hver gang repræsentantCheckbox funktionen kaldes.

Indholdet i repræsentantCheckbox funktionen er stort set identisk med suppleantCheckbox. I repræsentantCheckbox-funktionen berøres hovedsageligt indhold relateret til repræsentant-feltet. Den samme input-validering er gældende. Her bliver den globale variabel repChecked sat, ligesom supChecked blev sat i suppleantCheckbox-funktionen.

### **Funktion checkLength(input, maxLength, field)<sup>17</sup>**

Denne valideringsmetode kontrollerer, at input-længden opfylder det maksimalt tilladte for det pågældende felt. Hvis valideringen opfyldes, så returneres der sandt. Hvis input-valideringen ikke opfyldes, modtager brugeren en fejlbesked fra systemet, hvor brugeren får oplyst det pågældende felt, samt hvad den maximale længde på feltet må være, og der returneres falsk.

### **Funktion isString(input, field)<sup>18</sup>**

Denne valideringsmetode kontrollerer, at input udelukkende består af en af karakterne a-z, A-Z, æøå, ÆØÅ samt bindestreg. Dette udtryk er defineret som en regular expression  $^[a-zA-ZæøåÆØÅ-]+\$$ .

Hvis valideringen opfyldes, returneres der sandt. Hvis input-valideringen ikke opfyldes, modtager brugeren en fejlbesked fra systemet, hvor brugeren får oplyst feltnavnet og fejlbesked, og der returneres falsk.

### **Funktion isNumeric(input, field)<sup>19</sup>**

Denne valideringsmetode tjekker at input udelukkende består af tal. Dette udtryk er defineret som en regular expression  $^[0-9]+\$$ .

Hvis valideringen opfyldes, returneres der sandt. Hvis input-valideringen ikke opfyldes, modtager brugeren en fejlbesked fra systemet, hvor brugeren får oplyst feltnavnet og fejlbesked, og der returneres falsk.

---

<sup>16</sup> User\_management.php, linie: 128-152

<sup>17</sup> User\_management.php, linie: 157-165

<sup>18</sup> User\_management.php, linie: 167-176

<sup>19</sup> User\_management.php, linie: 178-187

### **Funktion isEmpty(input, field)<sup>20</sup>**

Denne valideringsmetode kontrollerer, at længden på input er større end 0.

Hvis valideringen opfyldes, returneres der sandt. Hvis input-valideringen ikke opfyldes, modtager brugeren en fejlbesked fra systemet, hvor brugeren får oplyst feltnavnet og fejlbesked, og der returneres falsk.

### **Funktion isAlphaNumeric(input, field)<sup>21</sup>**

Denne valideringsmetode kontrollerer, at input består af bogstaver, tal, komma, punktum og bindestreg. Dette udtryk er defineret som en regular expression `^[a-zA-Z æøå ÆØÅ 0-9 , . -]+`.

Hvis valideringen opfyldes, returneres der sandt. Hvis input-valideringen ikke opfyldes, modtager brugeren en fejlbesked fra systemet, hvor brugeren får oplyst feltnavnet og fejlbesked, og der returneres falsk.

### **Funktion isEmail(input, field)<sup>22</sup>**

Denne valideringsmetode kontrollerer, at input har form som en e-mail. Dette udtryk er defineret som en regular expression `^[w\-\.\+]+\@[a-zA-Z0-9\-\.\+]\.[a-zA-Z0-9]{2,4}`.

Hvis valideringen opfyldes, returneres der sandt. Hvis input valideringen ikke opfyldes, modtager brugeren en fejlbesked fra systemet, hvor brugeren får oplyst feltnavnet og fejlbesked, og der returneres falsk.

### **Funktion isDate(input, field)<sup>23</sup>**

Denne valideringsmetode kontrollerer, at input har dato formen YYYY-MM-DD bestående af 4 tal bindestreg 2 tal bindestreg 2 tal. Dette udtryk er defineret som en regular expression `^(d{4})-(d{2})-(d{2})`.

Hvis valideringen opfyldes, returneres der sandt. Hvis input-valideringen ikke opfyldes, modtager brugeren en fejlbesked fra systemet, hvor brugeren får oplyst feltnavnet og fejlbesked, og der returneres falsk.

### **Funktion createValidate()<sup>24</sup>**

Denne funktion benyttes i forbindelse med input-valideringen i opret person-feltet. I opret person-feltet skal der udfyldes en række felter med navn, adresse, postnummer, telefon, mobil, e-mail, suppleantvalg, suppleantprioritet samt repræsentantvalg. Alle disse felter har hver deres egen validering, som sikrer, at dataene er korrekte.

Først hentes alle felternes værdier, og disse gemmes i lokale variabler. Derefter foretages validering på hver gemt lokal variabel ved at kalde de forskellige valideringsmetoder. Derefter køres der validering på hvert

---

<sup>20</sup> User\_management.php, linie: 190-198

<sup>21</sup> User\_management.php, linie: 200-209

<sup>22</sup> User\_management.php, linie: 211-220

<sup>23</sup> User\_management.php, linie: 222-231

<sup>24</sup> User\_management.php, linie: 233-286

enkelt felt, og ved valideringsfejl rettes fokus mod det pågældende element, og der returneres falsk. Der skal returneres sand for hele funktionen, før dataene videresendes til server-siden adduser.php

Suppleant- og repræsentantfelterne er afhængige af, om checkboxen er sand eller ej. Der udføres først et tjek, hvorved der undersøges, om den globale variabel supChecked er sand, samt om den globale variabel repChecked er sand.

Derefter kan selve input-valideringen finde sted på sup valg, sup prio og rep valg

Tabellen nedenunder viser sammenhængen mellem hvert felt og dets valideringsmetoder.

Felt	Validering
Navn	isString(name, "Navn") og checkLength(name, 30, "Navn")
Adresse	isAlphaNumeric(adress, "Adresse") og checkLength(adress, 40, "Adresse")
Postnummer	isNumeric(pnr, "Postnummer") og checkLength(address, 12, "Postnummer")
Tlf	isNumeric(tlf, "Tlf") og checkLength(tlf, 12, "Tlf")
Mobil	isNumeric(mobil, "Mobil") og checkLength(mobil, 12, "Mobil")
Email	isEmail(email, "E-mail") og checkLength(email, 30, "E-mail")
Sup valg	isNumeric(svalg, "Sup Valg") og checkLength(svalg, 4, "Sup Valg")
Sup prioritet	isNumeric(sprio, "Sup Prio") og checkLength(sprio, 4, "Sup Prio")
Rep Valg	isNumeric(rvalg, "Rep Valg") og checkLength(rvalg, 4, "Rep Valg")

Table 3 - Input validering - Opret Person

### Funktion addsupValidate()<sup>25</sup>

Denne funktion benyttes i forbindelse med input-valideringen af opdater medlemskab-feltet. Før selve input-valideringen finder sted, kontrolleres det, om personen allerede er medlem af repræsentantskabet eller bestyrelsen. Er dette sandt, modtager brugeren en fejlmeddelelse fra systemet. Hvis personen ikke er medlem af repræsentantskabet og bestyrelsen, hentes de indtastede suppleantvalgs- og suppleantprioritetsværdier.

Derefter køres validering på hvert enkelt felt, og ved valideringsfejl rettes fokus mod det pågældende element, og der returneres falsk. Der skal returneres sand for hele funktionen, før dataene videresendes til serversiden addsup.php

<sup>25</sup> User\_management.php, linie: 288-306

Sup valg	isNumeric(add_sup_valg, "Sup Valg") og checkLength(svalg, 4, "Sup Valg")
Sup prioritet	isNumeric(add_sup_prio, "Sup Prio") og checkLength(add_sup_prio, 4, "Sup Prio")

Tabel 4 - Input Validering - Opret Suppleant

### Funktion addrepValidate()<sup>26</sup>

Denne funktion benyttes i forbindelse med input-valideringen af opdater medlemskab-feltet. Før selve input-valideringen finder sted, kontrolleres det, om personen allerede er medlem af suppleantgruppen. Er dette sandt, modtager brugeren en fejlmeddelelse fra systemet. Hvis personen ikke er medlem af suppleantgruppen, hentes det indtastede repræsentantvalgs- og repræsentantsuppleantværdier.

Derefter køres validering på repræsentantvalg, og ved valideringsfejl rettes fokus mod det pågældende element, og der returneres falsk. Der skal returneres sand for hele funktionen, før dataene videresendes til serversiden addrep.php

Rep Valg	isNumeric(add_rep_valg, "Rep Valg") og checkLength(add_rep_valg, 4, "Rep Valg")
----------	---

Tabel 5 - Input Validering - Opret Repræsentant

### Funktion addbesValidate()<sup>27</sup>

Denne funktion benyttes i forbindelse med input-valideringen af opdater medlemskab-feltet. Før selve input-valideringen finder sted, kontrolleres det, om personen er medlem af suppleantgruppen og ikke medlem af repræsentantskabet. Er dette sandt, modtager brugeren en fejlmeddelelse fra systemet.

Hvis personen allerede er medlem af repræsentantskabet og ikke medlem af suppleanterne, hentes de indtastede bestyrelsesvalgs- og bestyrelsestitelværdier. Derefter køres der validering på bestyrelsesvalg og bestyrelsestitel, og ved fejl rettes fokus mod det pågældende element, og der returneres falsk. Der skal returneres sandt for hele funktionen, før dataene videresendes til serversiden addbes.php

Bes Valg	isDate(add_bes_valg, "Bes Valg")
Bes Titel	Check om add_bes_titel er tom

Tabel 6 - Input Validering - Tilføj Til Bestyrelsen

<sup>26</sup> User\_management.php, linie: 308-321

<sup>27</sup> User\_management.php, linie: 323-342

## Funktion delUserValidate()<sup>28</sup>

Denne funktion benyttes i forbindelse med input-valideringen i slet person-feltet. I slet person-feltet skal der vælges en person. Først hentes personvalget, der gemmes i en lokal variabel. Herefter kontrolleres det, om input er tomt. Ved tomt input rettes fokus mod vælg person-feltet i slet person, og der returneres falsk.

## Funktion editValidate()<sup>29</sup>

Denne funktion benyttes i forbindelse med input-valideringen i opdater person-feltet. I opdater person-feltet kan der redigeres i en række felter med navn, adresse, postnummer, telefon, mobil, e-mail, suppleantvalg, suppleantprioritet, repræsentantvalg, repræsentantprioritet, bestyrelsesvalg samt bestyrelsestitel. Alle disse felter har hver deres egen validering, som sikrer, at dataene er korrekte. Felterne, som omhandler suppleant, repræsentant og bestyrelse, vises kun til personer, som er medlem af den pågældende gruppe. Hvis personen er medlem af suppleantgruppen, vil suppleantfelterne være synlige. Er personen medlem af repræsentantgruppen, vil de pågældende repræsentantfelter være synlige. Er personen medlem af bestyrelsesgruppen, vil bestyrelsesfelterne være synlige.

Først hentes alle felternes værdier, og disse gemmes i lokale variabler. Derefter køres validering på hvert enkelt felt, og der køres ligeledes validering på de synlige gruppefelter (suppleant, repræsentant og bestyrelse). Ved valideringsfejl rettes fokus mod det pågældende element, og der returneres falsk. Der skal returneres sand for hele funktionen, før dataene videresendes til serversiden ediusers.php

Felt	Validering
Navn	isString(edi_name, "Navn") og checkLength(edi_name, 30, "Navn")
Adresse	isAlphaNumeric(edi_adr, "Adresse") og checkLength(edi_adr, 40, "Adresse")
Postnummer	isNumeric(edi_pnr, "Postnummer") og checkLength(edi_pnr, 12, "Postnummer")
Tlf	isNumeric(edi_tlf, "Tlf") og checkLength(edi_tlf, 12, "Tlf")
Mobil	isNumeric(edi_mob, "Mobil") og checkLength(edi_mob, 12, "Mobil")
Email	isEmail(edi_email, "E-mail") og checkLength(edi_email, 30, "E-mail")
Sup valg	isNumeric(edi_svalg, "Sup Valg") og checkLength(edi_svalg, 4, "Sup Valg")
Sup prioritet	isNumeric(edi_sprio, "Sup Prio") og checkLength(edi_sprio, 4, "Sup Prio")
Rep Valg	isNumeric(edi_rvalg, "Rep Valg") og checkLength(edi_rvalg, 4, "Rep Valg")
Bes Valg	isDate(edi_bvalg, "Bes Valg") og check at edi_btitel ikke er tom

Tabel 7 - Input Validering - Opdater Person

<sup>28</sup> User\_management.php, linie: 344-352

<sup>29</sup> User\_management.php, linie: 354-424



### Funktion showAdmin(str)<sup>30</sup>

Denne funktion benyttes i forbindelse med feltet opdater admin. I opdater admin-feltet skal man vælge en person fra select-listen, hvorefter javascript-metoden showAdmin(aid) kaldes og tager admin-id som input-parameter. Metoden kaldes, hver gang select-listen ændres, og showAdmin funktionen kaldes med det pågældende admin-id. Der er ydermere tilføjet et <div id='admin'> felt, som opdateres med data, hver gang showAdmin kaldes.

Indholdet i showAdmin funktion er magen til indholdet i showMember. Den eneste forskel er i xmlhttp.open, hvor forespørgslen til webserveren sættes til getadmin.php?q=aid.

### Funktion createAdminValidate()<sup>31</sup>

Denne funktion benyttes i forbindelse med input-valideringen i opret admin-feltet. I opret admin-feltet skal der udfyldes en række felter med brugernavn, e-mail samt kodeord. Alle disse felter har hver deres egen validering, som sikrer at dataene er korrekte.

Først hentes alle felternes værdier, og de gemmes i lokale variabler. Derefter laves der validering på hver gemt lokal variabel ved at kalde de forskellige valideringsmetoder. Derefter køres der validering på hvert enkelt felt, og ved valideringsfejl rettes fokus mod det pågældende element, og der returneres falsk. Der skal returneres sand for hele funktionen, før dataene videresendes til serversiden addadmin.php

Tabellen nedenunder viser sammenhængen mellem hvert felt og dets valideringsmetoder.

Felt	Validering
Brugernavn	isAlphaNumeric(adminusr, "Admin Brugernavn") og checkLength(adminusr, 30, "Admin Brugernavn")
E-mail	isEmail(adminemail, "Admin Email") og checkLength(adminemail, 30, "Admin Email")
Kodeord	Adminpass længden må ikke være under 8

Tabel 8 - Input Validering - Opret Admin

### Funktion delAdminValidate()<sup>32</sup>

Denne funktion benyttes i forbindelse med input-valideringen i slet admin-feltet. I slet admin-feltet skal der vælges en admin. Først hentes admin-valget, og det gemmes i en lokal variabel. Herefter kontrolleres det, om input er tomt. Ved tomt input rettes fokus mod vælg admin-feltet i slet admin, og der returneres falsk.

<sup>30</sup> Admin\_management linie: 10-34

<sup>31</sup> Admin\_management linie: 80-101

<sup>32</sup> Admin\_management linie: 102-110

## Funktion editAdminValidate()<sup>33</sup>

Denne funktion benyttes i forbindelse med input-valideringen i rediger admin-feltet. I rediger admin-feltet kan der redigeres i en række felter såsom admin brugernavn, admin e-mail samt admin kodeord. Alle disse felter har hver deres egen validering, som sikrer at dataene er korrekte.

Først hentes alle felternes værdier, og de gemmes i lokale variabler. Derefter køres der validering på hvert enkelt felt, og ved valideringsfejl rettes fokus mod det pågældende element, og der returneres falsk. Der skal returneres sandt for hele funktionen, før dataene videresendes til serversiden ediadmin.php

Felt	Validering
Brugernavn	isAlphaNumeric(edi_austr, "Admin Brugernavn") og checkLength(edi_austr, 30, "Admin Brugernavn")
E-mail	isEmail(edi_aema, "Admin Email") og checkLength(edi_aema, 30, "Admin Email")
Kodeord	Edi_apass længden må ikke være under 8

Tabel 9 - Input Validering - Rediger Admin

## Backend – Serverfiler

Der er en række ligheder i alle backend-filer. Først startes en php-session, derefter kontrolleres det, om sessionen er gyldig og har rettigheder til at foretage ændringer i denne fil. Er dette ikke tilfældet, sendes brugeren til login-siden, og en fejlbesked gemmes i en session-variabel 'error', som kan se fejlbeskeden på login.php siden. Hvis sessionen er gyldig, kan der rettes forespørgsler til databasen.

### Fil - adduser.php

Denne fil kaldes efter javascript-funktionen createValidate() har returneret sandt. Hvis sessionen er gyldig, hentes alle input-værdier fra opret person-felterne med navn, adresse, postnummer, telefon, mobil, e-mail, r-siden, sommerhusadresse, sommerhustelefon, sommerhuspostnummer, repræsentantvalg, repræsentantprioritet, suppleantvalg samt suppleantprioritet.

Systemet finder ud af hvilke af disse felter der er udfyldt, og derefter sendes der forespørgsler til de forskellige databasetabeller.

Først sendes der en INSERT-forespørgsel til persontabellen med data såsom navn, adresse, postnummer, telefon, mobil og e-mail. Denne forespørgsel får tildelt et person-id-nummer (pid), som benyttes ved alle de øvrige forespørgsler.

<sup>33</sup> Admin\_management linie: 112-132

Derefter sendes der en INSERT-forespørgsel til sommerhus-tabellen med data som pid, sommerhusadresse, sommerhuspostnummer samt sommerhustelefon. Denne forespørgsel får tildelt et sommerhus-id-nummer (sid), som benyttes ved nogle af de andre forespørgsler.

Hvis repræsentantfeltet er udfyldt, sendes der en INSERT-forespørgsel til repræsentant-tabellen med data som pid, sid, valgår, sup. Sid kan være null eller et numerisk tal. Derfor er der udarbejdet to forskellige INSERT-forespørgsler til repræsentant-tabellen, hvoraf den ene indeholder sid som null og den anden sid som det pågældende sommerhus-id. Dette er baseret på, om personen, som er oprettet, har udfyldt de 3 sommerhusfelter. Hvis de tre sommerhusfelter er udfyldt, er personen ikke-fastboende og får tildelt sid til et numerisk tal. Hvis de tre sommerhusfelter ikke er udfyldt, er personen fastboende og får tildelt sid til null.

Hvis suppleantfeltet er udfyldt, sendes der en INSERT-forespørgsel til suppleant-tabellen med data såsom pid, sid, suppleant valg år samt suppleant prioritet. Samme opdeling er også gældende her, hvad angår fastboende sommerhusejere og ikke-fastboende sommerhusejere.

Derefter indsættes der et standardbillede i image-tabellen. Der køres en række php-funktioner på standardbilledet, som sikrer, at der kan køres en INSERT-forespørgsel i image-tabellen med data som pid, filnavn, filtype, filstørrelse samt fildata.

Derefter bliver brugeren viderestillet til user\_management-siden ved hjælp af javascript-funktionen window.location.

Forespørgsler	Værdi felter	Bemærkning
INSERT – person	Navn, adresse, postnummer, telefon, mobil, e-mail, r-siden	Her findes pid-værdien, som hentes og gemmes til andre forespørgsler
INSERT - sommerhus	Pid, sommerhusadresse, sommerhuspostnummer, sommerhustelefon	Her findes sid-værdien, som hentes og gemmes til andre forespørgsler
INSERT – repræsentant	Pid, sid, valg, sup	Sid-værdien er enten null eller et numerisk tal
INSERT - suppleant	Pid, sid, valg, prioritet	Sid værdien er enten null eller et numerisk tal
INSERT - image	Pid, filnavn, filtype, filstørrelse, fildata	

Tabel 10 - Database Forespørgsler – Opret Person

## Fil – addadmin.php

Denne fil kaldes efter javascript funktionen createAdminValidate() har returneret sandt. Denne fil er stort set identisk med adduser.php-filen. Den eneste forskel består i, hvilke variabelværdier der hentes, og hvilke INSERT-forespørgsler der køres. Derfor bliver det kun vist i tabellen nedenfor.

Forespørgsel	Værdi felter	Bemærkning
INSERT – admin	Admin-brugernavn, admin-kodeord, admin-e-mail	

Tabel 11 - Database Forespørgsler - Opret Admin

## Fil – addsup.php

Denne fil benyttes i opdater medlemskab-feltet, når en person skal tilføjes som suppleant.

Først hentes alle input værdier via POST<sup>34</sup> fra opdater medlemskab-felterne såsom suppleant valg og suppleantprioritet. Person-id'et hentes fra URL'en ved hjælp af GET<sup>35</sup>.

Derefter tjekkes med databasen, om personen er fastboende sommerhusejer eller ikke-fastboende sommerhusejer.

Baseret på, om personen er fastboende eller ej, køres den rette INSERT-forespørgsel.

Forespørgsel	Værdi felter	Bemærkning
INSERT – suppleant	Pid, sid, valg, prioritet	Sid-værdien er enten null eller et numerisk tal baseret på, om personen er fastboende eller ej

Tabel 12 - Database Forespørgsler - Opret Suppleant

## Fil – addrep.php

Denne fil benyttes i opdater medlemskab-feltet, når en person skal tilføjes som repræsentant. Filen er stort set identisk med addsup.php. Den eneste forskel består i, hvilke variabelværdier der hentes, og hvilke INSERT-forespørgsler der køres. Derfor bliver det kun vist i tabellen nedenfor.

Forespørgsel	Værdi felter	Bemærkning
INSERT – repræsentant	Pid, sid, valg, sup	Sid værdien er enten null eller et numerisk tal baseret på, om personen er fastboende eller ej

Tabel 13 - Database Forespørgsler - Opret Repræsentant

## Fil – addbes.php

Denne fil benyttes i opdater medlemskab-feltet, når en person skal tilføjes som bestyrelsesmedlem. Den er stort set identisk med addsup.php. Den eneste forskel består i, hvilke variabelværdier der hentes, og hvilke INSERT-forespørgsler der køres. Derfor bliver det kun vist i tabellen nedenfor.

Forespørgsel	Værdi felter	Bemærkning
INSERT – bestyrelsen	Pid, valg, titel	

Tabel 14 - Database Forespørgsler - Tilføj Til Bestyrelsen

## Fil – addimg.php

Denne fil benyttes i personbilledfeltet, når en person skal have tilføjet et nyt billede. Først kontrolleres det, at et billede er uploadet, og at størrelsen på billedet er større end 0. Derefter hentes person-id'et pid via

<sup>34</sup> POST, hentes fra inputfeltet i den pågældende form

<sup>35</sup> GET, hentes fra URL'en

GET, hvorpå de forskellige billeddata såsom filnavn, midlertidig filnavn, filstørrelse og filtype hentes via POST.

Billeddataene bliver nu behandlet af en række php-funktioner med henblik på, at acceptable data kan indsættes i databasen. Når dataene er klar, køres der en INSERT-forespørgsel til image-tabellen. Tabellen nedenunder viser forespørgslen.

Forespørgsel	Værdi felter	Bemærkning
INSERT – image	Pid, fil navn, fil type, fil størrelse, fil data	

Tabel 15 - Database Forespørgsler - Tilføj Billede

### Fil – deluser.php

Denne fil benyttes i slet person-feltet, når en person skal slettes helt fra systemet. Personen vil blive slettet helt fra alle tabeller i databasen. Først hentes person-id'et via POST fra user\_management. Derefter køres der en række forespørgsler til databasen med det pågældende person-id for at se, i hvilke tabeller personen er oprettet. Tabellen nedenfor viser hvilke forespørgsler der eksekveres.

Tabel	Forespørgsel
Person	SELECT – person
Sommer	SELECT – sommerhus
Repræsentant	SELECT – repræsentant
Suppleant	SELECT – suppleant
Bestyrelsen	SELECT – bestyrelsen
Image	SELECT – image

Tabel 16 - Database Forespørgsler – Select

Nu ved systemet, hvilke tabeller personen er oprettet i, og derfor køres kun DELETE-forespørgsler på de tabeller, hvori personen optræder.

Tabel	Forespørgsel
Person	DELETE – person
Sommer	DELETE – sommerhus
Repræsentant	DELETE – repræsentant
Suppleant	DELETE – suppleant
Bestyrelsen	DELETE – bestyrelsen
Image	DELETE – image

Tabel 17 - Database Forespørgsler - Slet Person

### Fil – deladmin.php

Denne fil benyttes i slet admin-feltet, når en admin skal slettes helt fra systemet. Først hentes admin-id'et via POST fra admin\_management. Derefter køres der en forespørgsel til admin-tabellen i databasen for at kontrollere, om det pågældende admin-id findes i databasen.

Nu kan systemet slette admin-brugeren fra databasen ved at køre forespørgslen DELETE på admin-tabellen.

### Fil – delbes.php

Denne fil benyttes i opdater medlemskab-feltet, når en person skal slettes fra en bestyrelsespost. Denne fil fungerer på samme måde som deladmin.php, blot med den ændring, at personen fjernes fra bestyrelsestabellen i databasen.

### Fil – delsup.php

Denne fil benyttes i opdater medlemskab-feltet, når en person skal slettes fra suppleantgruppen. Denne fil fungerer på samme måde som deladmin.php, blot med den ændring, at personen fjernes fra suppleanttabellen i databasen.

### Fil – delrep.php

Denne fil benyttes i opdater medlemskab-feltet, når en person skal slettes fra repræsentantskabet. Denne fil fungerer på samme måde som deladmin.php, blot med den ændring, at personen fjernes fra repræsentanttabellen i databasen.

### Fil – delimg.php

Denne fil benyttes i person billeder-feltet, når et billede skal slettes for en given person. Denne fil fungerer på samme måde som deladmin.php, blot med den ændring, at billedet fjernes fra imagetabellen i databasen.

### Fil – ediuser.php

Denne fil benyttes i opdater person-feltet, når en person skal have redigeret data, som er relateret til personen. Først hentes person-id'et via POST fra user\_management. Derefter køres der en række forespørgsler til databasen med det pågældende person-id for at se, hvilke tabeller personen er oprettet i. Sommerhus-feltet vil altid være synligt, uanset om personen er fastboende eller ikke-fastboende. Det er derfor altid muligt at redigere i/tilføje til sommerhusinformationerne. Tabellen nedenfor viser hvilke forespørgsler der eksekveres.

Tabel	Forespørgsel
Person	SELECT – person
Sommer	SELECT – sommerhus
Repræsentant	SELECT – repræsentant
Suppleant	SELECT – suppleant
Bestyrelsen	SELECT – bestyrelsen
Image	SELECT – image

Tabel 18 - Database Forespørgsler - Select

Nu ved systemet, hvilke tabeller personen er oprettet i, og derfor køres kun UPDATE-forespørgsler på de tabeller, hvori personen optræder.

Den ene tabel, sommerhus, fungerer anderledes. Hvis personen er ikke-fastboende, og hvis alle 3 sommerhusfelter ændres, så de står tomme, opdateres suppleant- og repræsentant-tabellen, hvor sommerhus-

id'et sættes til null, og personen fjernes fra sommerhustabellen. Hvis alle 3 sommerhusfelter indeholder oplysninger, opdateres alle 3 sommerhusfelter med de nye informationer.

Hvis personen er fastboende, og personen udfylder alle tre sommerhusfelter, tilføjes personen i sommerhustabellen. Først hentes sommerhus-id'et fra sommerhustabellen. Derefter opdateres personens sommerhus-id (sid) i suppleant- og repræsentant-tabellerne til en numerisk værdi.

### **Fil – editadmin.php**

Denne fil benyttes i opdater admin-feltet, når en admin-bruger skal have redigeret i dets informationer. Først hentes admin-id'et via POST fra admin\_management. Derefter køres der en forespørgsel til databasen med det pågældende admin-id for at sikre, at id'et findes i admin-tabellen. Hvis id'et findes i admin-tabellen, hentes alle admin-felterne via POST, og derefter køres UPDATE-forespørgslen med det pågældende admin-id. Derved opdateres admin-brugerens informationer.

### **Fil – getuser.php**

Denne fil kaldes fra javascript funktionen showUser. showUser kalder denne fil med et id i URL'en. I getUser.php hentes id via GET og derefter køres der en række forespørgsler til databasen med det pågældende person-id for at se, hvilke tabeller personen er oprettet i. Formålet med denne php-fil er at returnere html-koden med relevante data til opdater person-feltet i user\_management. Dette sikrer, at når en person vælges, så er det kun muligt at redigere i de data, hvor personen er medlem. Hvis personen således ikke er suppleant, vil suppleantfeltet ikke kunne ses og dermed ikke kunne redigeres. Hvis personen er repræsentant, så vil repræsentantfeltet kunne ses og dermed redigeres. Det samme er gældende for bestyrelsen.

Hvis id'et findes i persontabellen, så returneres html samt data fra persontabellen.

Hvis id'et findes i sommerhustabellen, så returneres html samt data fra sommerhustabellen. Hvis id'et ikke findes i sommerhustabellen, så returneres html samt tom data (tomme inputfelter).

Hvis id'et findes i suppleant tabellen, så returneres html samt data fra suppleanttabellen.

Hvis id'et findes i repræsentanttabellen, så returneres html samt data fra repræsentanttabellen.

Hvis id'et findes i bestyrelsestabellen, så returneres html samt data fra bestyrelsestabellen.

### **Fil – getadmin.php**

Denne fil kaldes fra javascript funktionen showAdmin. ShowAdmin kalder denne fil med et id i URL'en. Denne fil har samme funktionalitet og er stort set identisk med getuser.php. Den eneste forskel består i, at der i getadmin.php returneres html og data fra admin-tabellen.

### **Fil – getmember.php**

Denne fil kaldes fra javascript funktionen showMember. ShowMember kalder denne fil med et id i URL'en. I getmember.php hentes id'et via GET, og derefter køres der en række forespørgsler til databasen med det pågældende person-id for at se, hvilke tabeller personen er oprettet i. Formålet med denne php-fil er at returnere html-kode med relevante data til opdater medlemskab-feltet i user\_management.

Hvis id'et findes i suppleanttabellen, returneres der html med tilhørende form og slet-knap. Hvis id'et ikke findes i suppleant tabellen, returneres der html med tilhørende input felter samt en tilføj-knap.

Hvis id'et findes i repræsentanttabellen, returneres der html med tilhørende html form, som er en fjern rep knap. Hvis id ikke findes i repræsentant tabellen, returneres der html med tilhørende input felter samt en tilføj knap.

Hvis id'et findes i bestyrelsestabelen, returneres der html med tilhørende html-form, som er en fjern bes-knap. Hvis id'et ikke findes i bestyrelsestabelen, returneres der html med tilhørende input-felter samt en tilføj-knap.

Dertil gælder der, hvis id'et findes i repræsentant tabellen, returneres der html med tilhørende html-form, som er en find-knap, der benyttes når en repræsentant eller et bestyrelsesmedlem erstattes med en suppleant.

Denne tabel viser sammenhængen mellem tabellerne, og hvilke html-former der returneres. Disse html-former indeholder 'action' som kalder andre php-filer. Derudover kan man se, hvilken knap der vises.

Tabel	Form action	Knap
Suppleant - id findes	Delsup.php?pid=\$q	Fjern sup
Suppleant - id findes ikke	Addsup.php?pid=\$q	Tilføj sup
Repræsentant – id findes	Delrep.php?pid=\$q	Fjern rep
Repræsentant – id findes ikke	Addrep.php?pid=\$q	Tilføj rep
Bestyrelsen – id findes	Delbes.php?pid=\$q	Fjern bes
Repræsentant – id findes	Find.php?pid=\$q	Find

Tabel 19 - Sammenhæng mellem tabel, form og knap for getmember.php

### Fil – getimg.php

Denne fil kaldes fra javascript funktionen showImg. ShowImg kalder denne fil med et id i URL'en. I getimg.php hentes id'et via GET, og derefter køres der en forespørgsel til databasen med det pågældende billed- id for at kontrollere, om id'et findes i imagetabellen. Formålet med denne php-fil er at returnere html-kode med relevant data til personbilledfeltet i user\_management.

Nedenstående tabel viser, hvilke html-former der returneres. Disse html-former indeholder 'action', som kalder andre php-filer. Derudover kan man se, hvilken knap der vises.

Tabel	Form action	Knap
Image – id findes	Delimg.php?pid=\$q	Fjern billede
Image – id findes ikke	Addimg.php?pid=\$q	Tilføj billede

Tabel 20 - Sammenhæng mellem tabel, form og knap for getimg.php



## Fil – find.php

Denne fil benyttes i opdater person-feltet, når en repræsentant erstattes af en suppleant. Først hentes person id'et via POST fra user\_management.

Derpå hentes personens repræsentant data, såsom sommerhus id og valg år. Herefter finder systemet ud af, om personen er fastboende sommerhusejer (sid IS NULL) eller ikke-fastboende sommerhusejer (sid IS NOT NULL).

Der findes en suppleant fra samme valgår, med samme sid og højeste prioritet. Hvis der ikke kan findes en suppleant fra samme valgår, findes en tilfældig suppleant (ORDER BY Rand()).

Prioriteten for suppleanterne fra samme valgår reduceres med 1. Suppleanten med prioritet 2 får således prioriteten 1.

Derefter indsættes den fundne suppleant i repræsentanttabellen med de pågældende data såsom pid, sid, valg år og sup.

Efterfølgende slettes den gamle repræsentant fra repræsentanttabellen. Den fundne suppleant slettes også fra suppleanttabellen og erstattes af en vakant person. Denne skal betragtes som en fiktiv person, hvor der er en ledig plads. Hvis den gamle repræsentant også er opført i bestyrelsestabelen, slettes vedkommende også fra denne tabel. Derefter indsættes den fundne suppleant også i bestyrelsestabelen med de samme data som forgænger.

## 9.5. Input validering

Der kan implementeres to typer input-validering, hvoraf det første er input-validering hos klient, og det andet er input-validering på serveren. Input-validering på serveren er en lang og ressourceomfattende proces, hvor klienten indtaster dataene, som sendes til serveren. På serveren valideres dataene, og ved fejl i valideringen sendes alle dataene tilbage til klienten. Dette er en langsom og krævende proces. Derimod kan input-validering på klientsiden udføres hurtigt og effektivt, da serveren ikke bliver overbelastet med validerings-metoder og kald til serveren. Input-validering på klientsiden kan gøres ved hjælp af javascript.

Ved brug af javascript input-validering på klientsiden sikres en hurtig form for input validering inden dataene sendes til serveren.

Der er implementeret input-valideringsmetoder i alle input-felter. Valideringen gør brug af en kombination af definerede regular expressions og almindelige tjek. Input-valideringen er implementeret i de to filer user\_management.php og admin\_management.php, hvor begge filer indeholder forskellige validerings-funktioner og regular expressions.

## Valideringsfunktioner

Disse funktioner validerer input, og returnerer enten falsk, hvis valideringen fejler, og sand, hvis valideringen accepteres.

Metodenavn	Metodebeskrivelse
isString	Tjekker, om input er af karaktererne a-z, A-Z, æøå, ÆØÅ samt bindestreg
isAlphaNumeric	Tjekker, om input består af en kombination af a-z, A-Z, æøå. ÆØÅ, 0-9, komma, bindestreg samt punktum
isNumeric	Tjekker, om input er af tallene 0-9
isEmail	Tjekker, om input har en e-mail form
isEmpty	Tjekker, om længden på input er større end 0
isDate	Tjekker, om input er af formen XXXX-XX-XX samt tal
checkLength	Tjekker, at input-længden ikke er større, end den må være, svarende til længden på datatypen i databasen

Tabel 21 - Input Validering - Metodebeskrivelse

## Opret person

Tabellen herunder viser, hvilke input-felter der valideres med hvilke valideringsmetoder. Alle inputfelterne i opret person-feltet valideres undtagen de 3 sommerhus-felter, da sommerhus-feltet godt må fremstå tomt.

Felt	Validering
Navn	isString og checkLength
Adresse	isAlphaNumeric og checkLength
Postnummer	isNumeric og checkLength
Telefon	isNumeric og checkLength
Mobil	isNumeric og checkLength
E-mail	isEmail og checkLength
Sup Valg	isNumeric og checkLength
Sup Prio	isNumeric og checkLength
Rep Valg	isNumeric og checkLength

Tabel 22 - Input Validering - Anvendte Metoder - Opret Person

## Opdater medlemskab

Tabellen herunder viser, hvilke input felter der valideres med hvilke valideringsmetoder. Alle inputfelterne i opdater medlemskab-feltet valideres.

Felt	Validering
Tilføj Suppleant valg	isNumeric og checkLength
Tilføj Suppleant prioritet	isNumeric og checkLength
Tilføj Repræsentant valg	isNumeric og checkLength
Tilføj bestyrelse valg	isDate
Tilføj bestyrelse titel	isEmpty

Tabel 23 - Input Validering - Anvendte Metoder - Opret Medlemskab

## Opdater person

Tabellen herunder viser, hvilke input felter der valideres med hvilke valideringsmetoder. Alle inputfelterne i opdater person-feltet valideres.

Felt	Validering
Rediger navn	isString og checkLength
Rediger adresse	isAlphaNumeric og checkLength
Rediger postnummer	isNumeric og checkLength
Rediger telefon	isNumeric og checkLength
Rediger mobil	isNumeric og checkLength
Rediger e-mail	isEmail og checkLength

Tabel 24 - Input Validering - Anvendte Metoder - Opdater Person

## Slet person

Tabellen herunder viser, hvilke input felter der valideres med hvilke valideringsmetoder. Alle inputfelterne i slet person-feltet valideres.

Felt	Validering
Slet person	isEmpty

Tabel 25 - Input Validering - Anvendte Metoder - Slet Person

## Opret Admin

Tabellen herunder viser, hvilke input-felter der valideres med hvilke valideringsmetoder. Alle inputfelterne i opret admin-feltet valideres.

Felt	Validering
Admin brugernavn	isAlphaNumeric
Admin e-mail	isEmail
Admin kodeord	Længden er større end 7

Tabel 26 - Input Validering - Anvendt Metoder - Opret Admin

## Slet Admin

Tabellen herunder viser, hvilke input felter der valideres med hvilke valideringsmetoder. Alle inputfelterne i slet admin-feltet valideres.

Felt	Validering
Slet admin	isEmpty

Tabel 27 - Input Validering - Anvendte Metoder – Slet Admin

## Rediger Admin

Tabellen herunder viser, hvilke input felter der valideres med hvilke valideringsmetoder. Alle inputfelterne i rediger admin-feltet valideres.

Felt	Validering
Rediger admin brugernavn	isAlphaNumeric
Rediger admin e-mail	isEmail
Rediger admin kodeord	Længden er større end 7

Tabel 28 - Input Validering - Anvendte Metoder - Rediger Admin

## 9.6. Sikkerhed

### SQL Injection

SQL injection er stort problem i mange websider, og det er derfor vigtigt at få det implementeret, så man undgår, at uvedkommende personer får adgang til systemet eller skader systemet. Derfor er der i php funktioner til at tage hånd om dette. Der er blevet gjort brug af stripslashes og mysql\_real\_escape\_string funktionerne i forholdsvis login.php og email.php

Nedenstående tabel giver en kort beskrivelse af hvad de to funktioner gør.

Funktion	Beskrivelse
Stripslashes(string)	Fjerner \ fra input strengen
Mysql_real_escape_string(string)	Undgår specielle karakterer i input strengen, såsom \x00, \n, \r, \, \', \", \x1a

Tabel 29 - SQL Injection funktioner

### Login

Der er implementeret login-funktionalitet i systemet. Der er oprettet to forskellige brugertyper, admin og sysadmin. Admin-brugeren har mulighed for at administrere personer i CRUD-modulet. Sysadmin-brugeren har mulighed for at håndtere personer og admin-brugere i CRUD-modulerne.

På de forskellige php-sider er der implementeret login-sikkerhed, som kontrollerer, at den pågældende session er gyldig og aktiv. Derved vil bestemte sider og indhold på de forskellige sider kun være tilgængelige for de sessioner, som er gyldige og aktive.

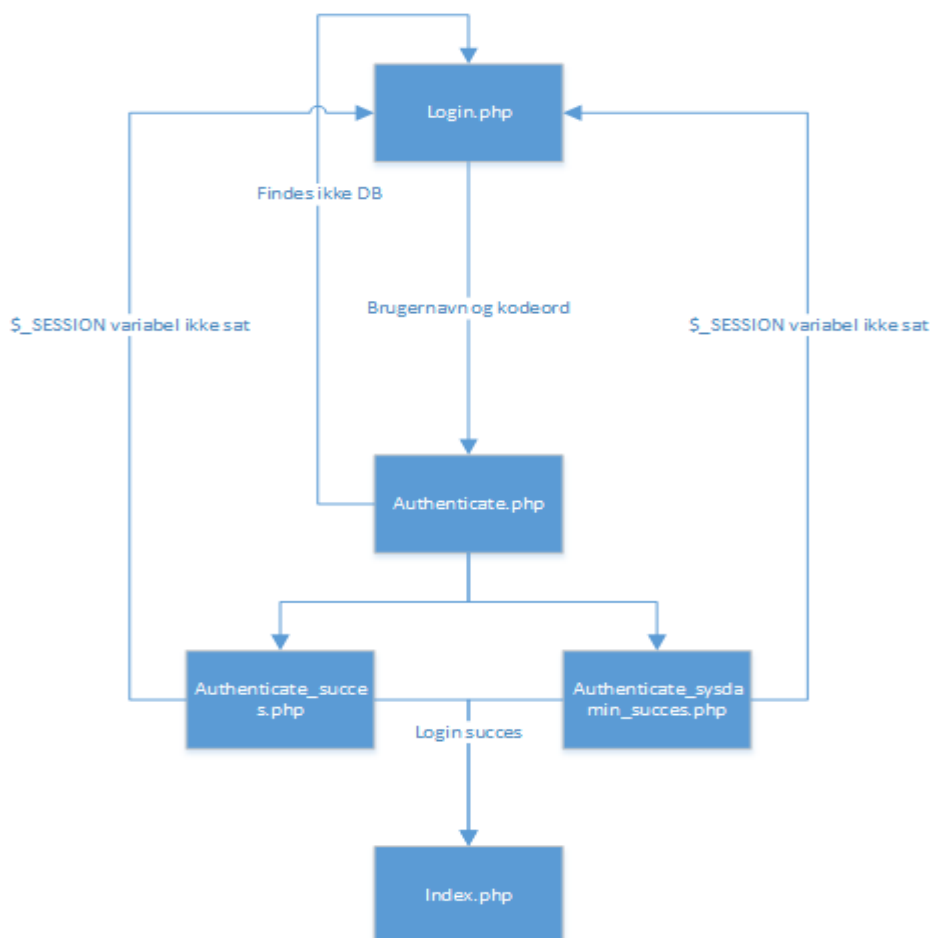
## Admin login

Der er implementeret et admin login, som sikrer, at kun admin-brugeren kan benytte CRUD modulet.

## Sys admin login

Der er implementeret et sys admin login, som sikrer, at man kan administrere admin-brugerne. Dette medfører, at der kan oprettes nye admin-brugere, at eksisterende admin brugere kan slettes, og at data for eksisterende admin brugere kan opdateres.

Nedenstående figur viser hvorledes flowet foregår ved login-funktionaliteten med de to forskellige brugere: admin og sysadmin.



Figur 10 – Flowchart – Login

## **login.php**

Denne side benyttes i forbindelse med login til systemet. Siden består af inputfelterne "brugernavn" og "kodeord" samt en submit- knap. Dette er lagt ned i en form som viderestilles til authenticate.php.

En bruger kan enten være admin-bruger eller sysadmin-bruger. Admin-brugeren har kun adgang til person-administrationsmodul, hvorimod sysadmin-brugeren har adgang til både personadministrationsmodul og admin-modul.

Derudover er der tilføjet et link "Glemte kodeord", som viderestiller til email.php.

## **authenticate.php**

Denne fil benyttes i forbindelse med autentificering af brugernavn og kodeord. Først startes en session, hvorefter der sikres imod Sql injection-angreb på databasen. Derefter hentes data fra databasen med det pågældende brugernavn og kodeord. Herefter undersøges det, om der findes et sæt af brugernavn og kodeord svarende til det, der blev modtaget fra login.php i enten admin-tabellen eller sysadmin-tabellen. Hvis det findes i admin tabellen, gemmes brugernavnet i `$_SESSION['username']`-variablen og der viderestilles til `authenticate_succes.php` siden. Hvis det findes i sysadmin tabellen, så gemmes brugernavnet både i `$_SESSION['username']` og `$_SESSION['sysadmin']` variablerne og der viderestilles til `authenticate_sysadmin_succes.php` siden.

Hvis der ikke findes et sæt af brugernavn og kodeord svarende til det, der blev modtaget fra login.php, gemmes en fejlbesked i `$_SESSION['error']`-variablen, og der viderestilles til login.php, hvor fejlbeskeden vil være synlig.

## **authenticate\_succes.php**

I denne fil tjekkes, at `$_SESSION['username']` er sat. Hvis det ikke er sat, viderestilles der til login.php. Hvis det er sat, så gøres `$_SESSION` variabelen sand, og der viderestilles til index.php

## **authenticate\_sysadmin\_succes.php**

I denne fil kontrolleres, at `$_SESSION['sysadmin']` er sat. Hvis det ikke er sat, viderestilles der til login.php. Hvis det er sat, gøres `$_SESSION` variabelen sand, og der viderestilles til index.php

## **checkLogin.php**

Dette er en global function, som benyttes på store dele af websiden. Denne funktion tjekker, om brugeren er logget ind med et gyldigt login, dvs. at `$_SESSION['username']` er sat, når brugeren navigerer rundt på de forskellige sider.

## **checkSysadminLogin.php**

Dette er en global function, som benyttes på store dele af websiden. Denne funktion tjekker, om brugeren er logget ind med et gyldigt login, dvs. at `$_SESSION['sysadmin']` er sat, når brugeren navigerer rundt på de forskellige sider.

## logout.php

Der viderestilles til denne side, når brugeren trykker på log ud i menuen. Denne side sørger for at ødelægge sessionen, og derefter viderestilles man til index.php

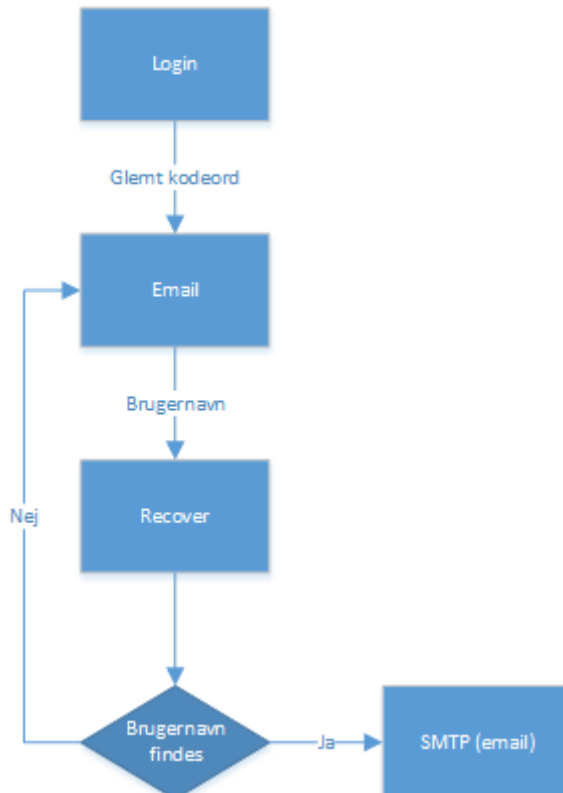
## email.php

Der viderestilles til denne side, når brugeren trykker på "glemt kodeord"-linket på login.php-siden. På denne side har brugeren mulighed for at indtaste sit brugernavn og få tilsendt login-informationer til den registrerede e-mailadresse. Når brugeren har indtastet brugernavn og valgt submit knappen, viderestilles der til recover.php-siden.

## recover.php

Brugernavnet fra email.php hentes via POST og sikres imod Sql injection-angreb på databasen. Derefter køres der en forespørgsel til admin-tabellen i databasen for at se, om brugernavnet findes i admin-tabellen. Hvis brugernavnet ikke findes i admin-tabellen, gemmes en fejlbesked i \$\_SESSION['error']-variablen, og der viderestilles til email.php-siden, hvor fejlbeskeden vil være synlig. Hvis brugernavnet findes i databasen, klargøres en e-mail med emnefelt, til e-mail, besked indeholdende brugernavn og kodeord. PHP's indbyggede mailfunktion gør brug af webserverens SMTP-server til at sende e-mailen af sted.

Nedenstående flowchart diagram visualiserer, hvilke processer admin-brugeren gennemgår fra start til slut, når pågældende har glemt sit kodeord og ønsker at få det sendt til den registrerede e-mailadresse.



Figur 11 - Flowchart - Glemt Login

## **Input validering**

Inputvalidering er implementeret vha. javascript. Javascript er et scriptsprog, som sikrer, at klienten kan interagere med klientside-scriptene, så man undgår at sende data frem og tilbage til serveren. Ved at implementere klientside-script er det den enkelte klient (browser), der overbelastes, fremfor serveren. Dermed bliver data først sendt til serveren, efter at inputvalideringen hos klienten er fuldført. Ved inputvalidering i serveren kan man risikere dårligere ydeevne, fordi der, når der er fejl i dataene, sendes data tilbage til klienten med fejlbeskeder. Ved inputvalidering i klienten får man en bedre ydeevne, fordi der, når der er fejl i dataene, ikke sendes data til serveren, idet data først sendes til serveren, når de er godkendte.

Inputvalidering sikrer en generel sikkerhed i systemet, da systemet derved bedre kan kontrollere de data, som lagres i databasen. Dette sikrer, at databasen ikke modtager fejlbehæftede data, eller data, som ikke er egnede til databasen.



# 10. Test

## 10.1. HTML validering

W3C (World Wide Web Consortium) tilbyder et valideringsværktøj, som sikrer, at websideudviklere kan teste den implementerede HTML eller XHTML og dermed sikre, at det er korrekt og uden fejl samt sikre at diverse browsere understøtter den implementerede HTML. Dette vil medføre en høj teknisk kvalitet af websiden.

Der er gennemført HTML valideringstests på alle websidens nedenfor anførte sider:

1. [www.rorvigvand.dk/index.php](http://www.rorvigvand.dk/index.php)
2. [www.rorvigvand.dk/kontaktpersoner.php](http://www.rorvigvand.dk/kontaktpersoner.php)
3. [www.rorvigvand.dk/webmaster.php](http://www.rorvigvand.dk/webmaster.php)
4. [www.rorvigvand.dk/bestyrelsen.php](http://www.rorvigvand.dk/bestyrelsen.php)
5. [www.rorvigvand.dk/medlemmer.php](http://www.rorvigvand.dk/medlemmer.php)
6. [www.rorvigvand.dk/suppleanter.php](http://www.rorvigvand.dk/suppleanter.php)
7. [www.rorvigvand.dk/repraesentantskab.php](http://www.rorvigvand.dk/repraesentantskab.php)
8. [www.rorvigvand.dk/historie.php](http://www.rorvigvand.dk/historie.php)
9. [www.rorvigvand.dk/login.php](http://www.rorvigvand.dk/login.php)
10. [www.rorvigvand.dk/email.php](http://www.rorvigvand.dk/email.php)
11. [www.rorvigvand.dk/meddelelser.php](http://www.rorvigvand.dk/meddelelser.php)
12. [www.rorvigvand.dk/moedeplan.php](http://www.rorvigvand.dk/moedeplan.php)
13. [www.rorvigvand.dk/sadmin.php](http://www.rorvigvand.dk/sadmin.php)
14. [www.rorvigvand.dk/sbestyrelsen.php](http://www.rorvigvand.dk/sbestyrelsen.php)
15. [www.rorvigvand.dk/sbilleder.php](http://www.rorvigvand.dk/sbilleder.php)
16. [www.rorvigvand.dk/sperson.php](http://www.rorvigvand.dk/sperson.php)
17. [www.rorvigvand.dk/srepraesentant.php](http://www.rorvigvand.dk/srepraesentant.php)
18. [www.rorvigvand.dk/ssommerhus.php](http://www.rorvigvand.dk/ssommerhus.php)
19. [www.rorvigvand.dk/ssuppleant.php](http://www.rorvigvand.dk/ssuppleant.php)
20. [www.rorvigvand.dk/admin\\_management.php](http://www.rorvigvand.dk/admin_management.php)
21. [www.rorvigvand.dk/user\\_management.php](http://www.rorvigvand.dk/user_management.php)

Nedenstående tabel viser testresultat og dokumentation for test for hver enkelt side.

Side	Resultat	Dokumentation
index.php	Valid	Figur 138
kontaktpersoner.php	Valid	Figur 139
webmaster.php	Valid	Figur 140
bestyrelsen.php	Valid	Figur 141
medlemmer.php	Valid	Figur 142
suppleanter.php	Valid	Figur 143
repraesentantskab.php	Valid	Figur 144
historie.php	Valid	Figur 145
login.php	Valid	Figur 146
email.php	Valid	Figur 147
meddelelser.php	Valid	Figur 148
moedplan.php	Valid	Figur 149
sadmin.php	Valid	Figur 150
sbestyrelsen.php	Valid	Figur 151
sbilleder.php	Valid	Figur 152
sperson.php	Valid	Figur 153
srepraesentant.php	Valid	Figur 154
ssommerhus.php	Valid	Figur 155
ssuppleant.php	Valid	Figur 156
admin_management.php	Valid	Figur 157
user_management.php	Valid	Figur 158

**Tabel 30 - HTML Valideringstests**

Baseret på testresultaterne fra W3C valideringsværktøjet, så kan der konkluderes at alle tilhørende sider er fejlfri og overholder HTML standarden.

## 10.2. CSS validering

W3C (World Wide Web Consortium) tilbyder et valideringsværktøj, som sikrer at websideudviklere kan teste den implementerede CSS og dermed sikre at det er korrekt og uden fejl samt at det understøttes af de største browsere, som er Internet Explorer, Mozilla Firefox, Google Chrome og Apple Safari. Dette vil medføre en høj teknisk kvalitet af websiden.

Der er gennemført CSS3 valideringstests på alle websidens sider herunder:

1. [www.rorvigvand.dk/index.php](http://www.rorvigvand.dk/index.php)
2. [www.rorvigvand.dk/kontaktpersoner.php](http://www.rorvigvand.dk/kontaktpersoner.php)
3. [www.rorvigvand.dk/webmaster.php](http://www.rorvigvand.dk/webmaster.php)
4. [www.rorvigvand.dk/bestyrelsen.php](http://www.rorvigvand.dk/bestyrelsen.php)
5. [www.rorvigvand.dk/medlemmer.php](http://www.rorvigvand.dk/medlemmer.php)
6. [www.rorvigvand.dk/suppleanter.php](http://www.rorvigvand.dk/suppleanter.php)
7. [www.rorvigvand.dk/repraesentantskab.php](http://www.rorvigvand.dk/repraesentantskab.php)
8. [www.rorvigvand.dk/historie.php](http://www.rorvigvand.dk/historie.php)
9. [www.rorvigvand.dk/login.php](http://www.rorvigvand.dk/login.php)
10. [www.rorvigvand.dk/email.php](http://www.rorvigvand.dk/email.php)
11. [www.rorvigvand.dk/meddelelser.php](http://www.rorvigvand.dk/meddelelser.php)
12. [www.rorvigvand.dk/moedeplan.php](http://www.rorvigvand.dk/moedeplan.php)
13. [www.rorvigvand.dk/sadmin.php](http://www.rorvigvand.dk/sadmin.php)
14. [www.rorvigvand.dk/sbestyrelsen.php](http://www.rorvigvand.dk/sbestyrelsen.php)
15. [www.rorvigvand.dk/sbilleder.php](http://www.rorvigvand.dk/sbilleder.php)
16. [www.rorvigvand.dk/sperson.php](http://www.rorvigvand.dk/sperson.php)
17. [www.rorvigvand.dk/srepraesentant.php](http://www.rorvigvand.dk/srepraesentant.php)
18. [www.rorvigvand.dk/ssommerhus.php](http://www.rorvigvand.dk/ssommerhus.php)
19. [www.rorvigvand.dk/ssuppleant.php](http://www.rorvigvand.dk/ssuppleant.php)
20. [www.rorvigvand.dk/admin\\_management.php](http://www.rorvigvand.dk/admin_management.php)
21. [www.rorvigvand.dk/user\\_management.php](http://www.rorvigvand.dk/user_management.php)

Nedenstående tabel viser testresultat og dokumentation for test for hver enkelt side.

Side	Resultat	Dokumentation
index.php	Valid	Figur 159
kontaktpersoner.php	Valid	Figur 160
webmaster.php	Valid	Figur 161
bestyrelsen.php	Valid	Figur 162
medlemmer.php	Valid	Figur 163
suppleanter.php	Valid	Figur 164
repraesentantskab.php	Valid	Figur 165
historie.php	Valid	Figur 166
login.php	Valid	Figur 167
email.php	Valid	Figur 168
meddelelser.php	Valid	Figur 169
moedeplan.php	Valid	Figur 170
sadmin.php	Valid	Figur 171
sbestyrelsen.php	Valid	Figur 172
sbilleder.php	Valid	Figur 173
sperson.php	Valid	Figur 174
srepraesentant.php	Valid	Figur 175
ssommerhus.php	Valid	Figur 176
ssuppleant.php	Valid	Figur 177
admin_management.php	Valid	Figur 178
user_management.php	Valid	Figur 179

Tabel 31 - CSS Valideringstests

### 10.3. Browser test

Dette afsnit beskriver hvorledes de forskellige platforme og browsere understøtter denne webside. Der er lavet forskellige tests, hvor selve HTML, CSS og funktionaliteten er afprøvet. Der er udført tests for de mest anvendte browsere samt mest anvendte platforme.

HTML test omhandler hvorledes de forskellige dele i websiden er repræsenteret, om alle sidens indhold bliver vist korrekt

CSS-test omhandler, hvorledes menuen fungerer, visning af billeder, visning af baggrundsbilleder, opdeling af siden, størrelsen på layout, farver, tekst, tekststørrelser osv.

Funktionalitet-test omhandler, hvorledes login-funktionaliteten fungerer, login-sessions, JavaScript-metoder, inputvalidering samt fejlbeskeder som benyttes i user management samt samspillet mellem klienten og serveren.

Platform	Browser	XHTML	CSS3	Funktionalitet	Screenshot
Windows XP	Internet Explorer 8.0	OK	OK	OK	Figur 180
Windows XP	Mozilla Firefox 20	OK	OK	OK	Figur 181
Windows XP	Google Chrome 26	OK	OK	OK	Figur 182
Windows XP	Apple Safari 5.1	OK	OK	OK	Figur 183
Windows 7	Internet Explorer 10	OK	OK	OK	Figur 184
Windows 7	Mozilla Firefox 20	OK	OK	OK	Figur 185
Windows 7	Google Chrome 26	OK	OK	OK	Figur 186
Windows 7	Apple Safari 5.1	OK	OK	OK	Figur 187
Windows 8	Internet Explorer 10	OK	OK	OK	Figur 188
Windows 8	Mozilla Firefox 20	OK	OK	OK	Figur 189
Windows 8	Google Chrome 26	OK	OK	OK	Figur 190
Windows 8	Apple Safari 5.1	OK	OK	OK	Figur 191
OSX Lion	Mozilla Firefox 20	OK	OK	OK	Figur 192
OSX Lion	Google Chrome 26	OK	OK	OK	Figur 193
OSX Lion	Apple Safari 6.0	OK	OK	OK	Figur 194
iOS6	iPad3 Safari	OK	OK	OK	Figur 195
iOS6	iPhone5 Safari	OK	OK	OK	Figur 196
Android 4.1	Samsung Galaxy SIII	OK	OK	OK	Figur 197
Android 4.1	Samsung Galaxy Note 10.1	OK	OK	OK	Figur 198

Tabel 32 – Browsertests

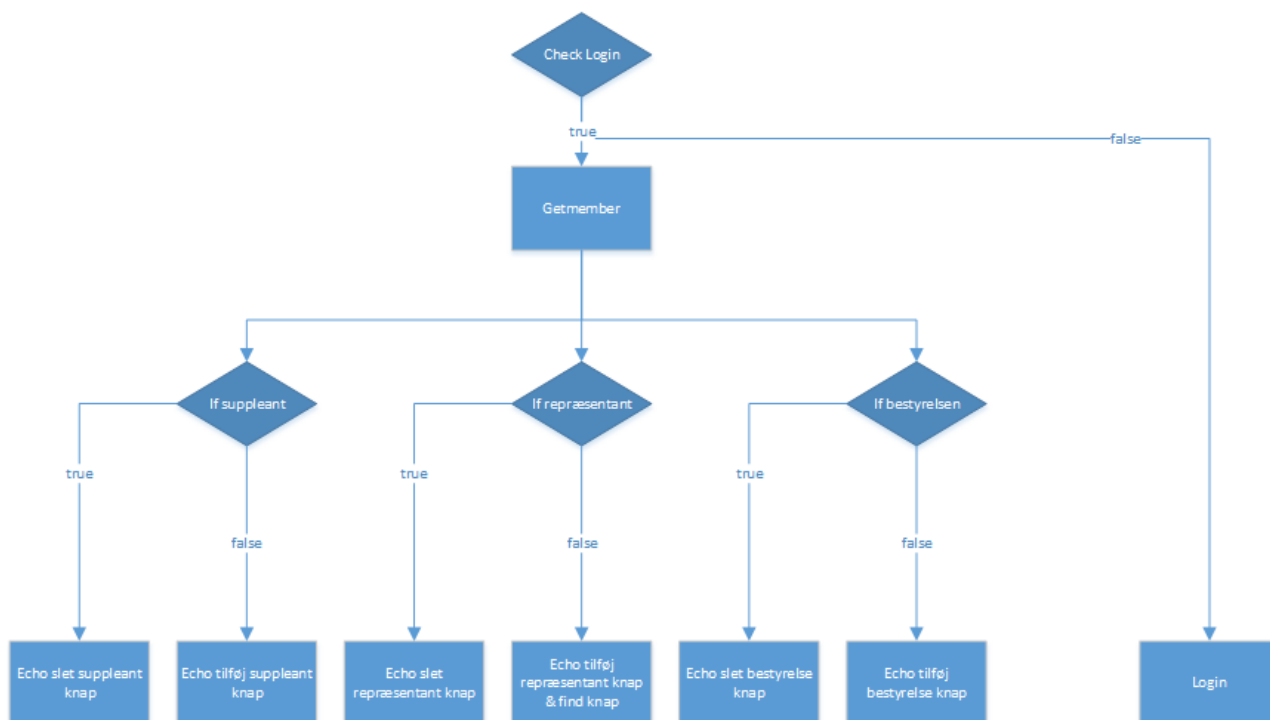
## 10.4. Whitebox test

Dette afsnit omhandler whitebox-test af et enkelt modul. Whitebox tests er bedst egnet til systemer med lange komplekse program stier, dvs. programmer med mange afhængigheder samt programmer med mange erklæringer<sup>36</sup> som kan have mange udfald. Derfor er det besluttet at dokumentere for udfaldet af en enkelt whitebox-test, da den generelle struktur, dvs. kodeopbygning, er stort set ens samt udfaldene på de forskellige erklæringer ikke er så store, da der typisk opereres med 3-6 forskellige udfald i hver fil. Denne test kan nærmere kategoriseres som en test af forgreningsdækningen<sup>37</sup>.

Nedenstående figur viser hvilke udfald, som er mulige i getmember filen. Som det kan ses, så er der 3 mulige udfald, som alle er dækket ind i if/else sætninger.

<sup>36</sup> Statement (for/while/if/else)

<sup>37</sup> Branch coverage



Figur 12 - Whitebox test - Getmember

## 10.5. Blackbox test

Dette afsnit omhandler blackbox-test af administrationsfunktionaliteten, login-funktionaliteten og inputvalideringen.

### Person administration

Nedenstående test-datasæt er til oprettelse af en person som ikke er fastboende.

Person -Test data 1	
Felter	Input
Navn	test
Adresse	testvej 1
Postnummer	2000
Tlf	12345678
Mobil	12345678
E-mail	test@test.dk
Sommerhus adr	Provevej 1
Sommerhus postnr	2800
Sommerhus tlf	12345678

Tabel 33 - Person test data 1

Nedenstående test-datasæt er til oprettelse af en person som er fastboende.

<b>Person - Test data 2</b>	
<b>Felter</b>	<b>Input</b>
Navn	test
Adresse	testvej 1
Postnummer	2000
Tlf	12345678
Mobil	12345678
E-mail	test@test.dk

Tabel 34 - Person test data 2

Nedenstående test-datasæt er tilknyttet en almindelig person som er fastboende.

<b>Person - Test data 3</b>	
<b>Felter</b>	<b>Input</b>
Navn	Andet navn
Adresse	andetvej 1
Postnummer	2300
Tlf	87654321
Mobil	87654321
E-mail	andet@andet.dk

Tabel 35 - Person test data 3

Nedenstående test-datasæt er tilknyttet en persons sommerhusinformationer.

<b>Sommerhus - Test data 1</b>	
<b>Felter</b>	<b>Input</b>
Sommerhus adr	Sommervej 1
Sommerhus postnr	2850
Sommerhus tlf	87654321

Tabel 36 - Sommerhus test data 1

Nedenstående test-datasæt er til oprettelse af en suppleant med forlængelse af datasættet fra enten person - test data 1 eller person - test data 2. Hvis suppleanten skal oprettes som ikke fastboende, benyttes person - test data 1. Hvis suppleanten skal oprettes som fastboende, benyttes person - test data 2.

<b>Suppleant – Test data 1</b>	
<b>Felter</b>	<b>Input</b>
Suppleant valg år	2013
Suppleant prioritet	1

Tabel 37 - Suppleant test data 1

<b>Suppleant – Test data 2</b>	
<b>Felter</b>	<b>Input</b>
Suppleant valg år	2016
Suppleant prioritet	2

Tabel 38 - Suppleant test data 2

Nedenstående test datasæt er til oprettelse af en repræsentant med forlængelse af datasættet fra enten person – test data 1 eller person – test data 2. Hvis repræsentanten skal oprettes som ikke fastboende, benyttes person – test data 1. Hvis repræsentanten skal oprettes som fastboende, benyttes person – test data 2.

<b>Repræsentant – Test data 1</b>	
<b>Felter</b>	<b>Input</b>
Repræsentant valg år	2013
Rep sup_t_B	IKKE NØDVENDIG

Tabel 39 - Repræsentant test data 1

<b>Repræsentant – Test data 2</b>	
<b>Felter</b>	<b>Input</b>
Repræsentant valg år	2016
Rep sup_t_B	IKKE NØDVENDIG

Tabel 40 - Repræsentant test data 2

Nedenstående test datasæt er til tilføjelse af en repræsentant i bestyrelsen.

<b>Bestyrelsen – Test data 1</b>	
<b>Felter</b>	<b>Input</b>
Bestyrelses valg år	2013-01-01
Bestyrelses titel	bestyrelsesmedlem

Tabel 41 - Bestyrelsen test data 1

<b>Bestyrelsen – Test data 2</b>	
<b>Felter</b>	<b>Input</b>
Bestyrelses valg år	2016-12-12
Bestyrelses titel	formand

Tabel 42 - Bestyrelsen test data 2

Nedenstående test datasæt er til tilføjelse af billede lige efter oprettelse af en person.

<b>Billede – Test data 1</b>	
<b>Felter</b>	<b>Input</b>
Vælg fil/Upload	Test.jpg

Tabel 43 - Billede test data 1



Nedenstående tabel viser testudfaldene for hver test med metodenavn, input, forventet output, faktisk output samt figurhenvi-  
 sning.

Metode	Input	Forventet output	Faktisk output	Figur
Opret alm. Person uden fast adresse	Person -Test data 1	Korrekt oprettet	Korrekt oprettet	Figur 69
Opret alm. Person med fast adresse	Person -Test data 2	Korrekt oprettet	Korrekt oprettet	Figur 70
Opret suppleant uden fast adresse	Person – Test data 1 samt Suppleant – Test data 1	Korrekt oprettet	Korrekt oprettet	Figur 71
Opret suppleant med fast adresse	Person – Test data 2 samt Suppleant – Test data 1	Korrekt oprettet	Korrekt oprettet	Figur 72
Opret repræsentant uden fast adresse	Person – Test data 1 samt Repræsentant – Test data 1	Korrekt oprettet	Korrekt oprettet	Figur 73
Opret repræsentant med fast adresse	Person – Test data 2 samt Repræsentant – Test data 1	Korrekt oprettet	Korrekt oprettet	Figur 74
Tilføj billede til oprettet person	Billede – Test data 1	Korrekt oprettet	Korrekt oprettet	Figur 75
Slet person	Vælg test personen	Slettet	Slettet	Figur 77
Opdater medlemskab – Tilføj suppleant	Vælg test personen samt Indtast suppleant – test data 1	Tilføjet til suppleant	Tilføjet til suppleant	Figur 77 og 78
Opdater medlemskab – tilføj repræsentant	Vælg test personen samt Indtast repræsentant – test data 1	Tilføjet til repræsentant	Tilføjet til repræsentant	Figur 79 og 80
Opdater medlemskab – tilføj repræsentant til bestyrelsen	Vælg test personen samt Indtast bestyrelsen – test data 1	Tilføjet til bestyrelsen	Tilføjet til bestyrelsen	Figur 81 og 82
Opdater medlemskab – Fjern fra bestyrelsen	Vælg test personen samt vælg fjern bes	Fjernet fra bestyrelsen	Fjernet fra bestyrelsen	Figur 83
Opdater medlemskab – Fjern fra repræsentant	Vælg test personen samt vælg fjern rep	Fjernet fra repræsentantskab	Fjernet fra repræsentantskab	Figur 84
Opdater medlemskab – Fjern bestyrer fra repræsentant	Vælg test personen samt vælg fjern rep	Fjernet fra repræsentantskab og bestyrelsen	Fjernet fra repræsentantskab og bestyrelsen	Figur 85
Opdater medlemskab – Fjern fra suppleant	Vælg test personen samt vælg fjern sup	Fjernet fra suppleant	Fjernet fra suppleant	Figur 86

Metode	Input	Forventet output	Faktisk output	Figur
Opdater medlemskab – Find (repræsentant)	Vælg test personen samt vælg find	Test repræsentant erstattet af suppleant med samme valgår og samme kreds	Erstattet repræsentant med suppleant fra samme valgår og samme kreds	Figur 86 og 87 er før der trykkes på find knappen. Figur 88 og 89 er efter der er trykket på find knappen
Opdater medlemskab – Find (bestyrelsen)	Vælg test personen samt vælg find	Test bestyrelsesmedlem erstattet af suppleant med samme valgår og samme kreds	Erstattet bestyrelsesmedlem med suppleant fra samme valgår og samme kreds	Identisk med 86,87,88 og 89. Den eneste forskel er at her tilføjes den fundne suppleant også til bestyrelsen. Figur 90 og 91
Opdater bruger – Rediger person data	Vælg test personen og erstæt person data med person – test data 3	Opdateringer gennemføres	Opdateringer gennemføres	Figur 92
Opdater bruger – Rediger sommerhus data	Vælg test personen og erstæt sommerhus data med Sommerhus - Test data 1	Opdateringer gennemføres	Opdateringer gennemføres	Figur 93
Opdater bruger – Rediger suppleant data	Vælg test suppleanten og erstæt suppleant data med suppleant - Test data 2	Opdateringer gennemføres	Opdateringer gennemføres	Figur 94
Opdater bruger – Rediger repræsentant data	Vælg test repræsentanten og erstæt repræsentant data med repræsentant - Test data 2	Opdateringer gennemføres	Opdateringer gennemføres	Figur 95
Opdater bruger – Rediger bestyrelse data	Vælg test bestyrelsesmedlemmet og erstæt bestyrelses data med bestyrelse - Test data 2	Opdateringer gennemføres	Opdateringer gennemføres	Figur 96
Opdater bruger – Rediger sommerhus data	Vælg test personen med sommerhus – test data 1 og fjern alle sommerhus data	Opdateringer gennemføres. Test personen har nu fast sommerhusadresse	Opdateringer gennemføres. Test personen har nu fast sommerhusadresse	Figur 97
Opdater bruger – Rediger sommerhus data	Vælg test personen uden noget sommerhus data	Opdateringer gennemføres. Test personen har nu ikke fast sommerhus adresse	Opdateringer gennemføres. Test personen har nu ikke fast sommerhus adresse	Figur 98

Metode	Input	Forventet output	Faktisk output	Figur
Bruger billeder – Fjern	Vælg test personen og vælg "fjern" billede	Opdateringer gennemføres. Billede fjernes og standardbillede indsættes	Opdateringer gennemføres. Billede fjernes og standardbillede indsættes	Figur 99
Bruger billeder - Tilføj	Vælg test personen og vælg "tilføj" billede Billede – Test data 1	Opdateringer gennemføres. Nyt billede tilknyttes til personen	Opdateringer gennemføres. Nyt billede tilknyttes til personen	Figur 100

Tabel 44 - Blackbox test uden forventede fejl

Nedenstående tabel viser testudfaldene for hver test med metodenavn, input, forventet output, faktisk output samt figurhenvi- sning. Disse tests indeholder forventede fejl, dvs. hvor input-validering skal komme med en fejlbesked.

Metode	Input	Forvente output	Faktisk output	Figur
Opret person – både repræsentant og suppleant	Afkryds både re- præsentant og suppleant	Input validering advarer og fjerner begge afkrydsning- er. Max 1 afkrydsning er tilladt	Input validering advarer og fjerner begge afkrydsning- er. Max 1 afkrydsning er tilladt	Figur 101
Opret person – navn	Test1	Inputvalidering ad- varer. Tal er ikke muligt	Inputvalidering ad- varer. Tal er ikke muligt	Figur 102
Opret person – navn	Test Test Test Test Test Test Test Test	Inputvalidering ad- varer. Max længde er 30 karakterer	Inputvalidering ad- varer. Max længde er 30 karakterer	Figur 103
Opret person – adresse	Test Test Test Test Test Test Test Test Test Test 1	Inputvalidering ad- varer. Max længde er 40 karakterer	Inputvalidering ad- varer. Max længde er 40 karakterer	Figur 104
Opret person – postnummer	Test	Inputvalidering ad- varer. Bogstaver er ikke muligt	Inputvalidering ad- varer. Bogstaver er ikke muligt	Figur 105
Opret person – postnummer	1234567890000	Inputvalidering ad- varer. Max længde er 12 karakterer	Inputvalidering ad- varer. Max længde er 12 karakterer	Figur 106
Opret person – tlf	Test	Inputvalidering ad- varer. Bogstaver er ikke muligt	Inputvalidering ad- varer. Bogstaver er ikke muligt	Figur 107
Opret person - tlf	1234567890000	Inputvalidering ad- varer. Max længde er 12 karakterer	Inputvalidering ad- varer. Max længde er 12 karakterer	Figur 108
Opret person – mobil	Test	Inputvalidering ad- varer. Bogstaver er ikke muligt	Inputvalidering ad- varer. Bogstaver er ikke muligt	Figur 109

Metode	Input	Forventet output	Faktisk output	Figur
Opret person – Email	@test.dk	Inputvalidering advarer. Der mangler karakterer foran @	Inputvalidering advarer. Der mangler karakterer foran @	Figur 111
Opret person – Email	test@dk	Inputvalidering advarer. Der mangler karakterer efter punktum	Inputvalidering advarer. Der mangler karakterer efter punktum	Figur 112
Opret person – Email	Test.test.dk	Inputvalidering advarer. Der mangler @	Inputvalidering advarer. Der mangler @	Figur 113
Opret suppleant – Sup valg år	Test	Inputvalidering advarer. Bogstaver er ikke muligt	Inputvalidering advarer. Bogstaver er ikke muligt	Figur 114
Opret suppleant – Sup valg år	12345	Inputvalidering advarer. Max længde er 4 karakterer	Inputvalidering advarer. Max længde er 4 karakterer	Figur 115
Opret suppleant – Sup prioritet	Test	Inputvalidering advarer. Bogstaver er ikke muligt	Inputvalidering advarer. Bogstaver er ikke muligt	Figur 116
Opret suppleant – Sup prioritet	123	Inputvalidering advarer. Max længde er 2 karakterer	Inputvalidering advarer. Max længde er 2 karakterer	Figur 117
Opret repræsentant – Rep valg år	Test	Inputvalidering advarer. Bogstaver er ikke muligt	Inputvalidering advarer. Bogstaver er ikke muligt	Figur 118
Opret repræsentant – Rep valg år	12345	Inputvalidering advarer. Max længde er 4 karakterer	Inputvalidering advarer. Max længde er 4 karakterer	Figur 119
Slet person	Ingen brugere valgt	Inputvalidering advarer. Der skal vælges en person	Inputvalidering advarer. Der skal vælges en person	Figur 120
Opdater medlemskab – Suppleant tilføjes som repræsentant	En suppleant tilføjes også som repræsentant. Rep valgår udfyldes	Inputvalidering advarer. Person er allerede suppleant og kan derfor ikke blive repræsentant	Inputvalidering advarer. Person er allerede suppleant og kan derfor ikke blive repræsentant	Figur 121
Opdater medlemskab – Suppleant tilføjes som bestyrer	En suppleant tilføjes også som bestyrer. Valgår udfyldes 2013-01-01 og bestyrelsesmedlem vælges som titel	Inputvalidering advarer. Person er allerede suppleant og kan derfor ikke blive bestyrer	Inputvalidering advarer. Person er allerede suppleant og kan derfor ikke blive bestyrer	Figur 122
Opdater medlemskab – Repræsentant tilføjes som suppleant	En repræsentant tilføjes også som suppleant ved at udfylde valgår 2013 og prioritet 1	Inputvalidering advarer. En repræsentant kan ikke tilføjes som suppleant	Inputvalidering advarer. En repræsentant kan ikke tilføjes som suppleant	Figur 123

Metode	Input	Forventet output	Faktisk output	Figur
Opdater medlemskab – alm. person tilføjes som bestyrer	En almindelig person uden nogle medlemskaber tilføjes som bestyrer ved at udfylde valgår 2013-01-01 og bestyrelsesmedlem som titel.	Inputvalidering advarer. Person skal være repræsentant for at blive bestyrer	Inputvalidering advarer. Person skal være repræsentant for at blive bestyrer	Figur 124
Opdater medlemskab – repræsentant tilføjes som bestyrer	Bestyrelses valgår sættes til test/te/te og titlen sættes til "vælg en titel"	Inputvalidering advarer. Valgår skal være på formen YYYY-MM-DD og bestå af tal. Der skal vælges en titel.	Inputvalidering advarer. Valgår skal være på formen YYYY-MM-DD og bestå af tal. Der skal vælges en titel.	Figur 125 og 126

Tabel 45 - Blackbox test med forventede fejl

Alle inputvalideringsfunktioner fra opret person er også gældende for felterne i opdater bruger-felterne, og der indgår derfor ikke dokumentation for disse i rapporten.

Suppleant og repræsentant-felterne fra opret person er også gældende i opdater medlemskab-felterne, og der indgår derfor ikke dokumentation for disse i rapporten.

### Admin management

Disse blackbox-tests omhandler tests af administrationsfunktionaliteten af sysadmin-delen.

Nedenstående test-datasæt er til oprettelse af en admin-bruger.

Admin – Test data 1	
Felter	Input
Brugernavn	Admintest1
E-mail	AdminTest@test.dk
Kodeord	Topsecret1

Tabel 46 - Admin test data 1

Nedenstående test datasæt er til oprettelse af en admin bruger.

Admin – Test data 2	
Felter	Input
Brugernavn	Admintest2
E-mail	AdminTest2@test.dk
Kodeord	Topsecret2

Tabel 47 - Admin test data 2

Nedenstående tabel viser testudfaldene for hver test med metodenavn, input, forventet output, faktisk output samt figurhenvi­sing.

Metode	Input	Forventet output	Faktisk output	Figur
Opret admin	Admin – Test data 1	Oprettet admin	Oprettet admin	Figur 127
Slet admin	Admintest1	Slettet admin	Slettet admin	Figur 128
Rediger admin	Erstat alle data med Admin – Test data 2	Ændring af data svarende til Admin – Test data 2	Ændring af data svarende til Admin – Test data 2	Figur 129

Tabel 48 - Blackbox test uden forventede fejl

Opret admin – Brugernavn	Testbruger1 Testbruger1 Testbruger#	Inputvalidering advarer. Max længde på 30 samt tegn er ikke gyldige	Inputvalidering advarer. Max længde på 30 samt tegn er ikke gyldige	Figur 130
Opret admin – Email	@test.dk	Inputvalidering advarer. Der mangler karakterer foran @	Inputvalidering advarer. Der mangler karakterer foran @	Figur 131
Opret admin – Email	test@dk	Input validering advarer. Der mangler karakterer efter punktum	Input validering advarer. Der mangler karakterer efter punktum	Figur 132
Opret admin – Email	admin.test.dk	Input validering advarer. Der mangler @	Input validering advarer. Der mangler @	Figur 133
Opret admin – Kodeord	forkort	Inputvalidering advarer. Kodeord skal være mindst 8 karakterer langt	Inputvalidering advarer. Kodeord skal være mindst 8 karakterer langt	Figur 134

Tabel 49 - Blackbox test med forventede fejl

Alle input valideringsfunktioner fra opret admin er også gældende for felterne i rediger admin felterne, og der indgår derfor ikke dokumentation for disse i rapporten.

## Login funktionalitet

Disse blackbox tests omhandler tests af login-funktionaliteten. Der er lavet scenarier omkring admin-login og sysadmin-login.

Metode	Input	Forventet output	Faktisk output	Figur X
Admin login	Brugernavn: Ahmet Kodeord: hemmelig	Login sikret til kun personadministration	Login sikret til kun personadministration	Figur 135
Sysadmin login	Brugernavn: Sysadmin Kodeord: hemmelig1	Login sikret til både personadministration og admin-administration	Login sikret til både personadministration og admin-administration	Figur 136
Admin login	Brugernavn: Ahmet Kodeord: forkertkode	Login ikke givet. Forkert brugernavn og/eller kodeord	Login ikke givet. Forkert brugernavn og/eller kodeord	Figur 137
Sysadmin login	Brugernavn: Sysadmin Kodeord: Forkertkode1	Login ikke givet. Forkert brugernavn og/eller kodeord	Login ikke givet. Forkert brugernavn og/eller kodeord	Figur 138
Admin – glemt kodeord	Brugernavn: Ahmet	E-mail sendt til registreret e-mailadresse med kodeord	E-mail sendt til registreret e-mailadresse med kodeord	Figur 139
Admin – glemt kodeord	Brugernavn: NotAhmet	Brugernavn ikke fundet	Brugernavn ikke fundet	Figur 140

Tabel 50 - Blackbox test – Login

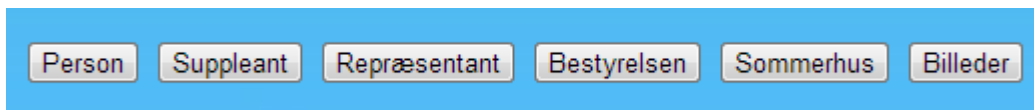
## 10.6. Konklusion på test

Der kan hermed konkluderes, at samtlige udførte testscenarier er gennemprøvet og, som forventet, har bestået prøven. Der er udført HTML-valideringstest, hvor alle testresultater er godkendt. Der er udført CSS-valideringstest, hvor alle testresultater er godkendt. Der er udført diverse browsertest, hvor alle testresultaterne er godkendt. Her kunne der udvides med flere platforme/browsere, men det er blevet undladt, da det blev besluttet kun at inddrage de mest populære platforme/browsere. Der er udført whitebox-test på et enkelt scenarie, hvor testresultatet er godkendt. Der er udført black box-test af person-administration, hvor alle testscenarier er godkendt. Her kunne man også udvide med flere testscenarier, men det er også undladt. Der er udført black box-test af admin-management, hvor alle testscenarier er godkendt. Alt i alt er systemet gennemtestet for fejl og kan dermed anvendes i produktion.

Alle testforsøg er dokumenterede og kan findes i bilagsafsnittet.

# 11. Brugervejledning

Se (admin)



Figur 13 - Brugervejledning - Se

Det første felt viser 6 knapper, som brugeren kan trykke på. Disse 6 knapper kan kun benyttes til at se diverse tabeller med data.

1. Klik på "person". Der vil blive viderestillet til en ny side, hvor der vil fremstå en liste over alle personer, som er registreret i systemet.
2. Klik på "suppleant". Der vil blive viderestillet til en ny side, hvor der vil fremstå en liste over alle suppleanter, som er registreret i systemet.
3. Klik på "repræsentant". Der vil blive viderestillet til en ny side, hvor der vil fremstå en liste over alle repræsentanter, som er registreret i systemet.
4. Klik på "bestyrelsen". Der vil blive viderestillet til en ny side, hvor der vil fremstå en liste over alle bestyrelsesmedlemmer, som er registreret i systemet.
5. Klik på "Sommerhus". Der vil blive viderestillet til en ny side, hvor der vil fremstå en liste over alle sommerhusejere, som er registreret i systemet.
6. Klik på "billeder". Der vil blive viderestillet til en ny side, hvor der vil fremstå en liste over alle personer samt deres tilknyttede billede.



## Opret almindelig Person (admin)

Suppleant:

Repræsentant:

Navn:

Adresse:

Postnummer:

Tlf.:

Mobil

E-mail:

R-siden:

Sommerhus Adr.:

Sommerhus Postnr.:

Sommerhus Tlf.:

Der er ikke valgt nogen fil

Figur 14 - Brugervejledning - Opret alm. person

Dette felt benyttes til at oprette en almindelig person i systemet. Her skal suppleant- eller repræsentantfeltet ikke afkrydses. Derefter skal felterne udfyldes med data:

1. Navn – her indtastes personens fulde navn.
2. Adresse – her indtastes personens faste adresse.
3. Postnummer – her indtastes postnummeret til adressen.
4. Tlf. – her indtastes personens fastnettelefonnummer.
5. Mobil – her indtastes personens mobiltelefonnummer.
6. E-mail – her indtastes personens e-mailadresse.
7. R-siden – her indtastes valgåret for, hvornår personen er meldt ind repræsentantskabet.
8. Sommerhus Adr. – her indtastes personens sommerhusadresse. Hvis dette felt efterlades tomt, registreres personen som fastboende.
9. Sommerhus Postnr. – her indtastes postnummeret til adressen. Hvis dette felt efterlades tomt, registreres personen som fastboende.
10. Sommerhus Tlf. – her indtastes telefonnummeret til sommerhuset. Hvis dette felt efterlades tomt, registreres personen som fastboende.
11. Sommerhus adr, sommerhus postnr. og sommerhus tlf. felterne skal alle 3 efterlades tomme for at registrere personen som fastboende.
12. Derefter klik på "tilføj".

13. Vælg fil – hvis man ønsker at tilføje et billede til personen, kan man vælge et billede lokalt fra pc'en.
14. Upload – billedet kan nu blive uploadet i systemet. Hvis der ikke tilføjes noget billede her, tilføjer systemet automatisk et standardbillede til personen.

### Opret suppleant (admin)

Suppleant:	<input checked="" type="checkbox"/>
Repræsentant:	<input type="checkbox"/>
Navn:	<input type="text"/>
Adresse:	<input type="text"/>
Postnummer:	<input type="text"/>
Tlf.:	<input type="text"/>
Mobil	<input type="text"/>
E-mail:	<input type="text"/>
R-siden:	<input type="text"/>
Sommerhus Adr.:	<input type="text"/>
Sommerhus Postnr.:	<input type="text"/>
Sommerhus Tlf.:	<input type="text"/>
Sup Valg år:	<input type="text"/>
Sup Prio:	<input type="text"/>

Der er ikke valgt nogen fil

Figur 15 - Brugervejledning - Opret suppleant

Dette felt benyttes til at oprette en suppleant i systemet. Her skal kun suppleantfeltet afkrydses. Der vil derefter blive vist 2 ekstra felter. Derefter skal felterne udfyldes med data:

1. Navn – her indtastes personens fulde navn.
2. Adresse – her indtastes personens faste adresse.
3. Postnummer – her indtastes postnummeret til adressen.
4. Tlf. – her indtastes personens fastnettelefonnummer.
5. Mobil – her indtastes personens mobiltelefonnummer.
6. E-mail – her indtastes personens e-mailadresse.
7. R-siden – her indtastes valgåret for, hvornår personen er meldt ind.
8. Sommerhus Adr. – her indtastes personens sommerhusadresse. Hvis dette felt efterlades tomt, registreres personen som fastboende.

9. Sommerhus Postnr. – her indtastes postnummeret til adressen. Hvis dette felt efterlades tomt, registreres personen som fastboende.
10. Sommerhus Tlf. – her indtastes telefonnummeret til sommerhuset. Hvis dette felt efterlades tomt, registreres personen som fastboende.
11. Sommerhus adr, sommerhus postnr. og sommerhus tlf. felterne skal alle 3 efterlades tomme for at registrere personen som fastboende.
12. Sup valg år – her indtastes valgåret for, hvornår personen er blevet suppleant.
13. Sup prio – her indtastes prioriteten for suppleanten. Prioriteten er typisk 1 eller 2.
14. Derefter klik på ”tilføj”.
15. Vælg fil – hvis man ønsker at tilføje et billede til personen, kan man vælge et billede lokalt fra pc'en.
16. Upload – Billedet kan nu blive uploadet i systemet. Hvis der ikke tilføjes noget billede her, så tilføjer systemet automatisk et standard billede til personen.

### Opret repræsentant (admin)

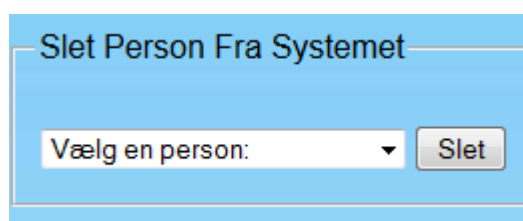
Suppleant:	<input type="checkbox"/>	
Repræsentant:	<input checked="" type="checkbox"/>	
Navn:	<input type="text"/>	
Adresse:	<input type="text"/>	
Postnummer:	<input type="text"/>	
Tlf.:	<input type="text"/>	
Mobil	<input type="text"/>	
E-mail:	<input type="text"/>	
R-siden:	<input type="text"/>	
Sommerhus Adr.:	<input type="text"/>	
Sommerhus Postnr.:	<input type="text"/>	
Sommerhus Tlf.:	<input type="text"/>	
Rep Valg år:	<input type="text"/>	
Rep Sup_t_B:	<input type="text"/>	
	<input type="button" value="Tilføj"/>	
<input type="button" value="Vælg fil"/>	Der er ikke valgt nogen fil	<input type="button" value="upload"/>

Figur 16 - Brugervejledning - Opret repræsentant

Dette felt benyttes til at oprette en repræsentant i systemet. Her skal kun repræsentantfeltet afkrydses. Der vil derefter blive vist 2 ekstra felter. Derefter skal felterne udfyldes med data:

1. Navn – her indtastes personens fulde navn.
2. Adresse – her indtastes personens faste adresse.
3. Postnummer – her indtastes postnummeret til adressen.
4. Tlf. – her indtastes personens fastnettelefonnummer.
5. Mobil – her indtastes personens mobiltelefonnummer.
6. E-mail – her indtastes personens e-mailadresse.
7. R-siden – her indtastes valgåret for, hvornår personen er meldt ind.
8. Sommerhus Adr. – her indtastes personens sommerhusadresse. Hvis dette felt efterlades tomt, registreres personen som fastboende.
9. Sommerhus Postnr. – her indtastes postnummeret til adressen. Hvis dette felt efterlades tomt, registreres personen som fastboende.
10. Sommerhus Tlf. – her indtastes telefon nummeret til sommerhuset. Hvis dette felt efterlades tomt, registreres personen som fastboende.
11. Sommerhus adr., sommerhus postnr. og sommerhus tlf. felterne skal alle 3 efterlades tomme for at registrere personen som fastboende.
12. Rep valg år – her indtastes valgåret for, hvornår personen er blevet repræsentant.
13. Rep Sup\_t\_B – udfyldelse af dette felt kan udelades, da feltet indikerer, at en suppleant er kommet i bestyrelsen.
14. Derefter klik på "tilføj".
15. Vælg fil – hvis man ønsker at tilføje et billede til personen, kan man vælge et JPEG-billede lokalt fra pc'en.
16. Upload – billedet kan nu blive uploadet i systemet. Hvis der ikke tilføjes noget billede her, så tilføjer systemet automatisk et standard billede til personen.

### Slet person (admin)



Figur 17 - Brugervejledning - Slet person

Dette felt benyttes til at slette personer fra systemet. Ved sletning af en person, vil denne blive fjernet fra hele systemet, dvs. at personen ikke længere vil fremstå i systemet:

1. Vælg person – ved at trykke på "pil ned" knappen vil en liste blive foldet ud med alle personer, som er registreret i systemet. Herefter skal man blot vælge en af personerne.
2. Klik på "slet" – Personen vil nu blive slettet fuldstændigt fra systemet.

## Opdater medlemskab (admin)

Dette felt benyttes til at opdatere medlemskabet for en person. Der findes 3 forskellige medlemskaber, suppleant, repræsentant og bestyrelse. Her vil man kunne tilføje/fjerne en person fra et af disse 3 medlemskaber.

1. Vælg person – ved at trykke på ”pil ned” knappen vil en liste blive foldet ud med alle personer, som er registreret i systemet. Herefter skal man blot vælge en af personerne.
2. Personen er ikke medlem – ved at udfylde et af de 3 felter kan personen tilmeldes den pågældende gruppe. For at blive bestyrelsesmedlem, så skal personen være medlem af repræsentantskabet.
3. Klik på ”tilføj” – de indtastede data vil nu blive tilknyttet den pågældende person, og personen vil være medlem af den pågældende gruppe.

Opdater Medlemskab

Ahmet Yildirim ▼

**Suppleant**  
Medlem: Nej

**Repræsentant**  
Medlem: Nej

**Bestyrelse**  
Medlem: Nej   ▼

Figur 18 - Brugervejledning - Opdater medlemskab - Tilføj

4. Suppleant – personen er medlem af suppleant gruppen og kan fjernes fra denne gruppe ved at trykke fjern.

Opdater Medlemskab

Ahmet Yildirim ▾

**Suppleant**  
 Medlem: Ja

**Repræsentant**  
 Medlem: Nej

**Bestyrelse**  
 Medlem: Nej   ▾

Figur 19 - Brugervejledning - Opdater medlemskab - Fjern

5. Repræsentant – personen er medlem af repræsentant gruppen og kan enten fjernes ved at trykke fjern eller erstattes af en anden suppleant ved trykke på find.

Opdater Medlemskab

Ahmet Yildirim ▾

**Suppleant**  
 Medlem: Nej

**Repræsentant**  
 Medlem: Ja

**Bestyrelse**  
 Medlem: Nej   ▾

**Erstat person**  
 Erstatte rep/bes:

Figur 20 - Brugervejledning - Fjern/Find repræsentant

6. Bestyrelsen – personen er medlem af bestyrelsen og kan enten fjernes ved at trykke fjern eller erstattes af en anden suppleant ved at trykke på find.

### Opdater Medlemskab

Ahmet Yildirim ▾

**Suppleant**  
 Medlem: Nej

**Repræsentant**  
 Medlem: Ja

**Bestyrelsen**  
 Medlem: Ja

**Erstat person**  
 Erstatte rep/bes:

Figur 21 - Brugervejledning - Fjern/find repræsentant eller fjern/find bestyrelsesmedlem

7. Hvis personen både optræder i bestyrelsen og repræsentantskabet, så vil fjern rep-knappen fjerne personen både fra repræsentantskabet og bestyrelsen. Dette er også gældende for find-knappen, hvor personen vil blive erstattet både i repræsentantskabet og bestyrelsen.

### Opdater person (admin)

Dette felt benyttes til at opdatere data for en person. Det vil være muligt at redigere i alle data, som personen er relateret til.

1. Vælg person – ved at trykke på "pil ned" knappen vil en liste blive foldet ud med alle personer, som er registreret i systemet. Herefter skal man blot vælge en af personerne.
2. Der vil blive vist en tabel med alle de felter, som kan redigeres. Der vil kun blive vist de input-felter som er relevante for personen. Dvs. at hvis personen sidder i bestyrelsen, vil man kunne ændre disse data. Hvis personen ikke sidder i suppleantgruppen, vil disse ikke blive vist.
3. Rediger de data, som skal ændres.
4. Klik på "gem" – data vil nu blive opdateret for personen.

Opdater Person

Ahmet Yildirim ▾

Person Data

Navn	Adresse	Postboks	Telefon	Mobil	E-mail	R-siden
Ahmet Yildirim	dtuvej 100C, vær 15	2800	12345678	87654321	ahmet@yildirim.dk	2013

Sommerhus Data

Sommerhus Adr	Sommerhus Postnr	Sommerhus Tlf
bjælken 3	2000	88888888

Repræsentant Data

Valg år	Sup_t_B
2013	

Bestyrelses Data

Valg	Titel
2013-01-12	Bestyrelsesmedlem ▾

Gem

Figur 22 - Brugervejledning - Opdater person

## Person billeder (admin)

Dette felt benyttes til at indsætte personbilleder. Her vil man kunne tilføje/fjerne et billede for en person.

1. Vælg person – ved at trykke på "pil ned" knappen vil en liste blive foldet ud med alle personer, som er registreret i systemet. Herefter skal man blot vælge en af personerne.
2. Vælg fil – hvis man ønsker at tilføje et billede til personen, kan man vælge et billede lokalt fra pc'en.
3. Upload – billedet kan nu blive uploadet i systemet. Hvis der ikke tilføjes noget billede her, tilføjer systemet automatisk et standardbillede til personen.

Person Billeder

Ahmet Yildirim ▾

Gennemse\_ upload

Figur 23 - Brugervejledning - Upload billede

4. Fjern – Personens billede kan ses og fjernes ved at klikke på "fjern".





Figur 24 - Brugervejledning - Fjern billede

### Login (admin)

Dette felt benyttes til admin-login og sysadmin-login. Herfra kan admin-brugeren logge ind med brugernavn og kodeord.

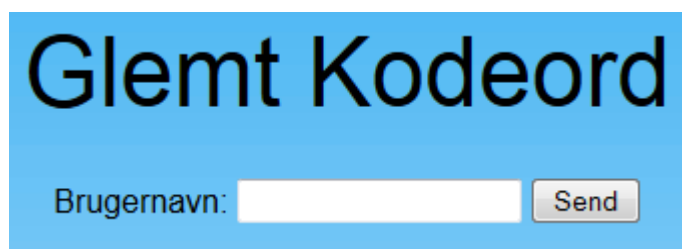
1. Brugernavn – indtast admin-brugernavnet
2. Kodeord – indtast kodeordet til det pågældende brugernavn
3. Klik på "Login" – det vil nu blive forsøgt at logge ind.

Figur 25 - Brugervejledning – Login

## Glemt Kodeord (admin)

Dette felt benyttes af admin-brugere, som har glemt deres kodeord. Ved dette tilfælde kan kodeordet sendes til admin-brugerens e-mailadresse.

1. Brugernavn – indtast admin-brugernavnet
2. Klik på "Send" – der vil nu blive sendt en e-mail med login-informationerne til den pågældende brugernavns e-mailadresse.

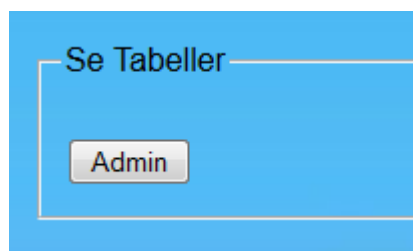


Figur 26 - Brugervejledning - Glemt kodeord

## Se (sysadmin)

Det første felt viser 1 knap, som sys admin-brugeren kan trykke på. Denne knap kan kun benyttes til at se data for de oprettede admin-brugere.

1. Klik på "admin". Der vil blive viderestillet til en ny side, hvor der vil fremstå en tabel over alle adminbrugere, som er registreret i systemet.

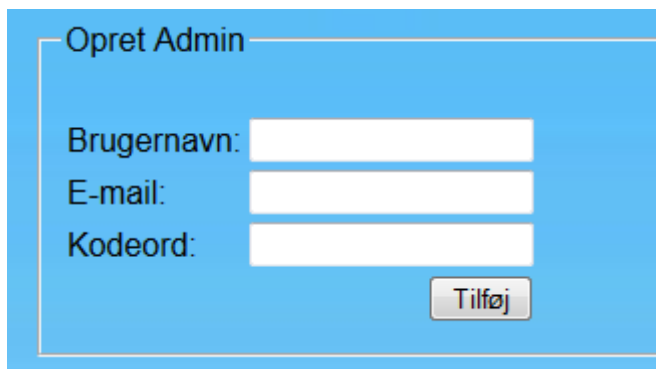


Figur 27 - Brugervejledning - Se admin

## Opret admin (sysadmin)

Dette felt benyttes til at oprette en admin-bruger i systemet. Derefter skal felterne udfyldes med data:

1. Indtast brugernavn – her skal admin-brugerens brugernavn stå. Brugernavnet er unikt, og det kan derfor kun være registreret én gang.
2. Indtast e-mail – her skal admin brugerens e-mailadresse stå. E-mailadressen er unik, og den kan derfor kun være registreret én gang.
3. Indtast kodeord – her skal admin-brugerens kodeord stå.
4. Klik på "Tilføj" – Admin-brugeren vil nu blive oprettet i systemet.

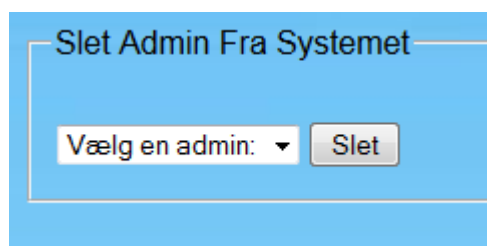


Figur 28 - Brugervejledning - Opret admin

### Slet admin (sysadmin)

Dette felt er til at slette admin-brugere fra systemet. Ved sletning af admin-brugeren vil brugeren blive fjernet fra hele systemet, dvs. at brugeren ikke længere vil fremstå i systemet:

1. Vælg admin – ved at trykke på ”pil ned” knappen, vil en liste blive foldet ud med alle admin-brugere, som er registreret i systemet. Herefter skal man blot vælge en af admin-brugerne.
2. Klik på ”slet” – admin-brugeren vil nu blive slettet fuldstændigt fra systemet.



Figur 29 - Brugervejledning - Slet admin

### Opdater admin (sysadmin)

Dette felt benyttes til at opdatere data for en admin-bruger. Det vil være muligt at redigere i alle data, som admin-brugeren har tilknyttet.

1. Vælg admin – ved at trykke på ”pil ned” knappen vil en liste blive foldet ud med alle admin-brugere, som er registreret i systemet. Herefter skal man blot vælge en af brugerne.
2. Der vil blive vist en tabel med alle de felter, hvori der kan redigeres.
3. Rediger de data, som skal ændres.

Klik på ”gem” – Dataene vil nu blive opdateret for personen.

Rediger Admin

Ahmet ▾

Admin Data

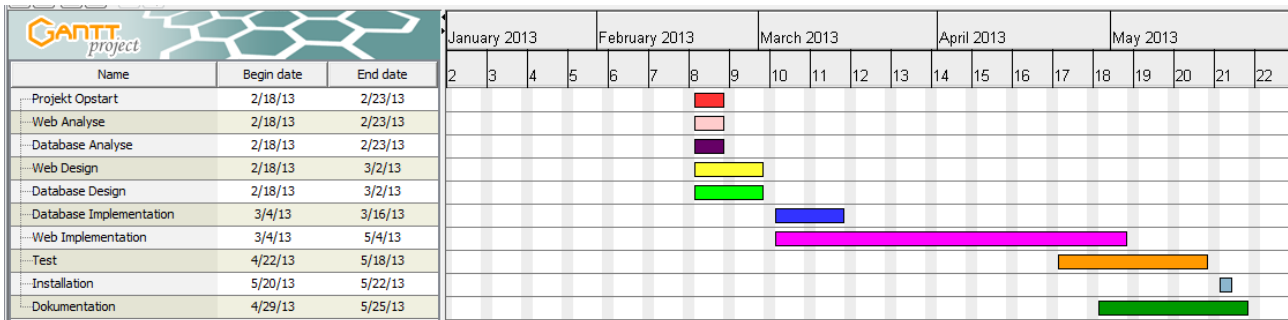
Brugernavn	Kodeord	E-mail
Ahmet	hemmelig	ahmet_yildirim89@hot

Gem

Figur 30 - Brugervejledning - Rediger admin

## 12. Projektplan

Der er udarbejdet en projektplan for dette projektforsløb med faser. For hver fase er der sat en startdato og en slutdato. Projektplanen er udarbejdet ved hjælp af programmet GanttProject. Denne projektplan er sat fra starten af projektet og har været med til at sætte deadlines for hvornår en milepæl skulle være nået.



Figur 31 - Gantt Projektplan

## 13. Installation

Alle PHP filer er udviklet lokalt i Netbeans, og derefter uploadet til roden af webserveren vha. FTP klienten filezilla. Databasen er også udviklet lokalt, og derefter eksporteret ned i en fil. Denne eksporterede fil er blevet importeret vha. PHPMYAdmin på webserveren.

## 14. Diskussion

### 14.1. Forbedringstiltag

Der er gjort en række tanker omkring ændringer eller forbedringer til den implementerede webside. En mulig ændring kunne være i form af ny teknologi, såsom HTML5, da det er den nye standard med forbedrede og nye muligheder.

En anden mulig ændring kunne vedrøre sikkerhedsaspektet i systemet, da der logges ind uden nogen form for sikkerhed, hvad angår andre tredjeparter. Der er ikke implementeret https/ssl, og derfor sendes brugernavn/ kodeord i klartekst, hvilket kan medføre, at andre tredjeparter kan udføre et Man-In-The-Middle-angreb og dermed opsnappe brugernavn og kodeord. Derudover kunne der også implementeres tidsstyrede sessioner for den bruger, som logger ind. Dvs. at hvis brugeren har været inaktiv på siden i fx 30 min, så logges der automatisk ud.

Admin- og sysadmin-brugernes kodeord lagres i databasen og dermed ikke krypteret med nogen form for krypteringsmetode. Her kunne der vælges at kryptere passwords med en md5 krypteringsalgoritme eller en sha1 krypteringsalgoritme.

En mulig ændring til selve websiden kunne være i form af mere CMS, dvs. dynamisk indhold fremfor statisk indhold. Dette kunne implementeres vha. moduler, hvor brugeren selv kunne bestemme, hvad der skulle stå, og hvilke billeder der skulle være på hver enkelt side. Der er adskillige CMS-systemer, som allerede tilbyder dette i form af Drupal, Wordpress osv. En anden mulighed ville være at implementere et CMS på egen hånd.

I forbindelse med upload af billeder for personerne sættes billederne til at blive vist på siden med en fast størrelse (højde x bredde), og dette bevirker, at personernes ansigter kan fremstå for smalle eller for brede. Derfor ville det være oplagt at implementere et automatisk billedredigeringsmodul, som automatisk gør billedet nøjagtigt så stort/småt, at det passer til en bestemt størrelse. En anden mulighed ville være at indsætte begrænsning, når brugeren forsøger at uploade et billede, dvs. at systemet kun kan acceptere billeder i en bestemt størrelse.

I forbindelse med logging af aktivitet, som admin-brugeren foretager på systemet, kunne der implementeres en log-funktionalitet, som nedskriver enhver aktivitet i en log-fil.

En ekstra feature til systemet ville være i form af, at afgåede repræsentanter bliver vist på historiesiden i kronologisk rækkefølge, således at websidens besøgende kan se, hvilke personer der har siddet i repræsentantskabet.

### 14.2. Metode

På grund af projektets omfang og manderessourcer så vil det være mindre passende at benytte agile software udviklingsmetoder som blandt andet scrum. I scrum har man fleksible deadlines, ændringer i kravene i gennem projektforløbet og ukendte krav fra starten af projektet. Ydermere arbejdes der i små hold, som regelmæssigt holder korte scrum møder omkring fremgangsmåde og strategi for næste sprint. Da der i dette projekt har været en enkelt mand til at imødekomme hele projektet, fremfor små hold bestående af X antal personer, så er der valgt ikke at benytte agile softwareudviklingsmetoder.

### 14.3. Vedligeholdelse

Systemet er udviklet efter de krav "kunden" havde, hvilket betyder at systemet ikke vil kunne klare enkelte modificeringer til den nuværende forretningsmodel. Det er vigtigt at forretningsmodellen bibeholdes intakt, så længe dette system ønskes. Ændringer i forretningsmodellen, som ikke vil gøre systemet ubrugeligt, er blandt andet antallet af repræsentanter, antallet af bestyrelsesmedlemmer og fast/ikke-fastboende sommerhusejere, billeder til personer, administrator brugere og system administrator brugere.

Systemet tåler ikke modificeringer til antallet suppleanter for hvert valg år. Det er implementeret specifikt at der skal være 2 suppleanter for hvert valg år. Disse 2 suppleanter skal forholdsvis have prioriteterne 1 og 2. I tilfælde af modificering af antallet af suppleanter for et givet valg år, så vil funktionaliteten på "find" knappen i person administrationsmodulet virke uhensigtsmæssigt. Funktionaliteten er beregnet til at kunne håndtere max 2 suppleanter for et givet valg år.



## 15. Konklusion

Dette projekt er udarbejdet over en tidsperiode på 12 uger, hvor formålet har været at refaktorere en gammel statisk webside til en nyere webside med dynamisk indhold. I den forbindelse er der designet og implementeret en helt ny webside i forbindelse med PHP, HTML 4.01 Transitional og CSS3. Ydermere er der implementeret adskillige javascript funktioner, som har til formål at gøre klientsiden mere dynamisk og brugervenlig. Databasen er også blevet refaktorert i den forbindelse, da der er tilføjet nye entiter med tilhørende attributter. Derudover er der udarbejdet ændringer til den gamle database, således at reglerne for 3. normalform er opfyldt.

Der er implementeret 2 CRUD moduler for henholdsvis admin-brugere og sysadmin-brugere. Det første modul har til formål at administrere personer i systemet: suppleanter, repræsentanter og bestyrelsesmedlemmer samt billeder af disse. Det andet modul har til formål at administrere admin-brugere. For at kunne differentiere mellem disse to brugertyper, er der implementeret et login-system, som muliggør, at admin-brugeren kun har adgang til personadministrationsmodulet, hvorimod sysadmin-brugeren har adgang til såvel personadministrationsmodulet som admin-administrationsmodulet.

Rørvig Vandværk-websiden har fået foretaget en fuldstændig renovering, herunder fået tilføjet 2 nye CRUD moduler. I den forbindelse har det første modul fået implementeret følgende funktioner: Opret person, slet person, opdater medlemskab, opdater person samt personbilleder. Det andet modul har fået implementeret følgende funktioner: Opret admin, slet admin samt rediger admin. Dertil er der implementeret input-valideringsfunktioner til alle felter ved hjælp af javascript.

Der er udført adskillige tests af de forskellige dele af systemet, herunder test af inputvalidering, whitebox-tests, blackbox- tests af funktionaliteten, browser-tests, HTML-valideringstests og CSS-valideringstests.

Den gamle webside<sup>38</sup> – som var en fuldstændig statisk webside – er hermed blevet renoveret og benytter på nuværende tidspunkt en nyere HTML-standard samt CSS i modsætning til det gamle system, som var implementeret vha. frames. I det gamle systemet blev bruger administrationen håndteret direkte i databasen, ved at opdatere tabellerne manuelt i databasen.

Det nye system indeholder en kombination af statisk og dynamisk indhold, hvilket nærværende projekt også omhandler. Systemet er installeret og er online på websiden [www.rorvigvand.dk](http://www.rorvigvand.dk) samt sat i produktion.

Gennem projektet er der anvendt en kombination af vandfaldsmodellen og spiralmodellen, hvilket har betydet at projektet er blevet faseopdelt i de 7 faser fra vandfaldsmodellen, hvorefter hver fase er blevet gennemgået en efter en. Der har ind i mellem været små iterationer bestående af analyse, design, implementation, integration, test og installation hvilket er karakteriseret ved spiralmodellen. Der er blevet udarbejdet en projektplan<sup>39</sup> fra start, som har været med til at muliggøre succeskriteriet, at tiden/deadlines overholdes. Dette har også medført et mere struktureret projektforsløb.

Derfor kan det konkluderes at projektet er udført og lever op til den stillede problemformulering.

---

<sup>38</sup> Figur 200

<sup>39</sup> Figur 31

## 16. Figurbetegnelser

Figur 1 - Sitemap - Brugere Uden Login.....	14
Figur 2 - Sitemap - Brugere Med Login.....	15
Figur 3 - Overordnet Arkitektur.....	17
Figur 4 - Conceptional Data Model.....	19
Figur 5 - Logical Data Model.....	20
Figur 6 - Physical Data Model.....	21
Figur 7 - Use Case - Main Use Case - Admin.....	24
Figur 8 - Main Use Case - Sysadmin.....	26
Figur 9 - Web Struktur.....	40
Figur 10 – Flowchart – Login.....	61
Figur 11 - Flowchart - Glemte Login.....	63
Figur 12 - Whitebox test - Getmember.....	70
Figur 13 - Brugervejledning - Se.....	80
Figur 14 - Brugervejledning - Opret alm. person.....	81
Figur 15 - Brugervejledning - Opret suppleant.....	82
Figur 16 - Brugervejledning - Opret repræsentant.....	83
Figur 17 - Brugervejledning - Slet person.....	84
Figur 18 - Brugervejledning - Opdater medlemskab - Tilføj.....	85
Figur 19 - Brugervejledning - Opdater medlemskab - Fjern.....	86
Figur 20 - Brugervejledning - Fjern/Find repræsentant.....	86
Figur 21 - Brugervejledning - Fjern/find repræsentant eller fjern/find bestyrelsesmedlem.....	87
Figur 22 - Brugervejledning - Opdater person.....	88
Figur 23 - Brugervejledning - Upload billede.....	88
Figur 24 - Brugervejledning - Fjern billede.....	89
Figur 25 - Brugervejledning – Login.....	89
Figur 26 - Brugervejledning - Glemte kodeord.....	90
Figur 27 - Brugervejledning - Se admin.....	90
Figur 28 - Brugervejledning - Opret admin.....	91
Figur 29 - Brugervejledning - Slet admin.....	91
Figur 30 - Brugervejledning - Rediger admin.....	92
Figur 31 - Gantt Projektplan.....	93
Figur 32 - Wireframe – Index.....	107
Figur 33 - Wireframe – Meddelelser.....	107
Figur 34 - Wireframe – Mødeplan.....	108
Figur 35 - Wireframe – Historie.....	108
Figur 36 - Wireframe – Kontaktpersoner.....	108
Figur 37 - Wireframe – Webmaster.....	109
Figur 38 - Wireframe – Medlemmer.....	109
Figur 39 - Wireframe – Bestyrelsen.....	110
Figur 40 - Wireframe – Suppleanter.....	110
Figur 41 - Wireframe - Person Administration.....	111

Figur 42 - Wireframe - Admin Administration.....	112
Figur 43 - Wireframe – Login.....	112
Figur 44 - Wireframe - Glemte Login.....	113
Figur 45 - Comps - Baggrundsbillede.....	113
Figur 46 - Use Case - Opret person.....	114
Figur 47 - Use Case - Slet person.....	114
Figur 48 - Use Case - Opdater Medlemskab.....	114
Figur 49 - Use Case - Opdater Person.....	115
Figur 50 - Use Case - Person billeder.....	115
Figur 51 - Flowchart - Opret Person.....	116
Figur 52 - Flowchart - Slet Person.....	117
Figur 53 - Flowchart - Opdater Medlemskab.....	118
Figur 54 - Flowchart - Opdater Person.....	119
Figur 55 - Flowchart - Person Billeder.....	120
Figur 56 - Flowchart - Opret Admin.....	121
Figur 57 - Flowchart - Slet Admin.....	122
Figur 58 - Flowchart - Rediger Admin.....	123
Figur 59 - Sekvensdiagram - Opret Person.....	124
Figur 60 - Sekvensdiagram - Slet Person.....	124
Figur 61 - Sekvensdiagram - Opdater Medlemskab.....	125
Figur 62 - Sekvensdiagram - Opdater Person.....	125
Figur 63 - Sekvensdiagram - Person Billeder.....	126
Figur 64 - Sekvensdiagram – Login.....	126
Figur 65 - Sekvensdiagram - Glemte Login.....	127
Figur 66 - Sekvensdiagram - Opret Admin.....	127
Figur 67 - Sekvensdiagram - Slet Admin.....	128
Figur 68 - Sekvensdiagram - Rediger Admin.....	128
Figur 69 - Blackbox test - Opret alm. Person.....	129
Figur 70 - Blackbox test – Person med fast adresse.....	129
Figur 71 - Blackbox test - Opret suppleant med fast adresse.....	129
Figur 72 - Blackbox test - Suppleant uden fast adresse.....	129
Figur 73 - Blackbox test – Opret repræsentant med fast adresse.....	129
Figur 74 - Blackbox test - Repræsentant uden fast adresse.....	129
Figur 75 - Blackbox test - Tilføj billede til person.....	129
Figur 76 - Blackbox test - Slet person.....	129
Figur 77 - Blackbox test - Opdater medlemskab - Tilføj suppleant.....	130
Figur 78 - Blackbox test - Opdater medlemskab - Tilføj suppleant DB.....	130
Figur 79 - Blackbox test - Opdater medlemskab - Tilføj repræsentant.....	130
Figur 80 - Blackbox test - Opdater medlemskab - Tilføj repræsentant DB.....	130
Figur 81 - Blackbox test - Opdater medlemskab - Tilføj til bestyrelsen.....	131
Figur 82 - Blackbox test - Opdater medlemskab - Tilføj til bestyrelsen DB.....	131
Figur 83 - Blackbox test - Fjern fra bestyrelsen.....	131
Figur 84 - Blackbox test - Fjern fra repræsentantskab.....	132

Figur 85 - Blackbox test - Fjern fra suppleant.....	132
Figur 86 - Blackbox test - Opdater medlemskab - Før find knappen trykkes i repræsentant tabel.....	132
Figur 87 - Blackbox test - Opdater medlemskab - Før find knappen trykkes i suppleant tabel.....	132
Figur 88 - Blackbox test - Opdater medlemskab - Efter find knappen trykkes i repræsentant tabel.....	132
Figur 89 - Blackbox test - Opdater medlemskab - Efter find knappen trykkes i suppleant tabel.....	132
Figur 90 - Blackbox test - Opdater medlemskab - Før find knappen trykkes i bestyrelsen.....	133
Figur 91 - Blackbox test - Opdater medlemskab - Efter find knappen trykkes i bestyrelsen.....	133
Figur 92 - Blackbox test - Opdater person.....	133
Figur 93 - Blackbox test - Opdater sommerhus.....	133
Figur 94 - Blackbox test - Opdater suppleant.....	133
Figur 95 - Blackbox test - Opdater repræsentant.....	133
Figur 96 - Blackbox test - Opdater bestyrelsen.....	133
Figur 97 - Blackbox test - Sommerhus tabel tom.....	133
Figur 98 - Blackbox test - Opdater sommerhus data.....	133
Figur 99 - Blackbox test - Fjern billede.....	134
Figur 100 - Blackbox test - Tilføj billede.....	134
Figur 101 - Blackbox test - Opret rep + sup.....	134
Figur 102 - Blackbox test - Navn skal være bogstaver.....	134
Figur 103 - Blackbox test - Navn max længde.....	134
Figur 104 - Blackbox test - Adresse max længde.....	135
Figur 105 - Blackbox test - Postnummer skal være tal.....	135
Figur 106 - Blackbox test - Postnummer max længde.....	135
Figur 107 - Blackbox test - Tlf skal være tal.....	136
Figur 108 - Blackbox test - Tlf max længde.....	136
Figur 109 - Blackbox test - Mobil skal være tal.....	136
Figur 110 - Blackbox test - Mobil max længde.....	137
Figur 111 - Blackbox test - Email ikke valid.....	137
Figur 112 - Blackbox test - Email ikke valid.....	137
Figur 113 - Blackbox test - Email ikke valid.....	138
Figur 114 - Blackbox test - Suppleant valg skal være tal.....	138
Figur 115 - Blackbox test – Suppleant valg max længde.....	139
Figur 116 - Blackbox test - Suppleant prioritet skal være tal.....	139
Figur 117 - Blackbox test - Suppleant prioritet max længde.....	140
Figur 118 - Blackbox test - Repræsentant valg skal være tal.....	140
Figur 119 - Blackbox test - Repræsentant valg max længde.....	141
Figur 120 - Blackbox test - Slet en person.....	141
Figur 121 - Blackbox test - Tilføj repræsentant.....	142
Figur 122 - Blackbox test - Tilføj til bestyrelsen.....	142
Figur 123 - Blackbox test - Tilføj til suppleant.....	143
Figur 124 - Blackbox test -Tilføj til bestyrelsen.....	143
Figur 125 - Blackbox test - Tilføj til bestyrelsen - Skal være i bestemt format.....	144
Figur 126 - Blackbox test - Tilføj til bestyrelsen - Vælg en titel.....	144
Figur 127 - Blackbox test - Opret admin.....	144

Figur 128 - Blackbox test - Slet admin .....	144
Figur 129 - Blackbox test - Rediger Admin .....	145
Figur 130 - Blackbox test - Brugernavn skal være bogstaver og tal .....	145
Figur 131 - Blackbox test - Brugernavn max længde .....	145
Figur 132 - Blackbox test - Email ikke valid.....	145
Figur 133 - Blackbox test - Email ikke valid.....	146
Figur 134 - Blackbox test - Email ikke valid.....	146
Figur 135 - Blackbox test - Kodeord minimum længde .....	146
Figur 136 - Blackbox test – Login .....	146
Figur 137 - Blackbox test - Glemte kodeord .....	147
Figur 138 - Blackbox test - Brugernavn ikke fundet .....	147
Figur 139 - HTML valideringstest - Index.....	147
Figur 140 - HTML valideringstest - Kontaktpersoner.....	147
Figur 141 - HTML valideringstest - Webmaster .....	148
Figur 142 - HTML valideringstest - Bestyrelsen .....	148
Figur 143 - HTML valideringstest - Medlemmer.....	148
Figur 144 - HTML valideringstest - Suppleanter .....	149
Figur 145 - HTML valideringstest - Repræsentantskab .....	149
Figur 146 - HTML valideringstest - Historie .....	149
Figur 147 - HTML valideringstest - Login .....	150
Figur 148 - HTML valideringstest - Email.....	150
Figur 149 - HTML valideringstest - Meddelelser.....	150
Figur 150 - HTML valideringstest - Mødeplan .....	151
Figur 151 - - HTML valideringstest - Sadmin.....	151
Figur 152 - HTML valideringstest - Sbestyrelsen .....	151
Figur 153 - HTML valideringstest - Sbilleder.....	152
Figur 154 - HTML valideringstest - Sperson.....	152
Figur 155 - HTML valideringstest - Srepræsentant.....	152
Figur 156 - HTML valideringstest - Ssommerhus.....	153
Figur 157 - HTML valideringstest - Ssuppleant.....	153
Figur 158 - HTML valideringstest - Admin management.....	153
Figur 159 - HTML valideringstest - User management.....	154
Figur 160 - CSS valideringstest - Index.....	154
Figur 161 - CSS valideringstest - Kontaktpersoner .....	154
Figur 162 - CSS valideringstest - Webmaster.....	154
Figur 163 - CSS valideringstest - Bestyrelsen.....	155
Figur 164 - CSS valideringstest - Medlemmer .....	155
Figur 165 - CSS valideringstest - Suppleanter.....	155
Figur 166 - CSS valideringstest - Repræsentantskab .....	155
Figur 167 - CSS valideringstest - Historie.....	155
Figur 168 - CSS valideringstest - Login.....	156
Figur 169 - CSS valideringstest - Email.....	156
Figur 170 - CSS valideringstest - Meddelelser .....	156

Figur 171 - CSS valideringstest - Mødeplan .....	156
Figur 172 - CSS valideringstest - Sadmin.....	156
Figur 173 - CSS valideringstest - Sbestyrelsen .....	157
Figur 174 - CSS valideringstest - Sbilleder .....	157
Figur 175 - CSS valideringstest - Sperson .....	157
Figur 176 - CSS valideringstest - Srepræsentant .....	157
Figur 177 - CSS valideringstest - Ssommerhus.....	157
Figur 178 - CSS valideringstest - Ssuppleant.....	158
Figur 179 - CSS valideringstest - Admin management .....	158
Figur 180 - CSS valideringstest - User management.....	158
Figur 181 - Browsertest - Win XP – IE 8.0 .....	159
Figur 182 - Browsertest - Win XP - Firefox 20.....	160
Figur 183 - Browsertest - Win XP - Chrome 26.....	161
Figur 184 - Browsertest - Win XP - Safari 5.1.....	162
Figur 185 - Browsertest - Win 7 - Explorer 10 .....	163
Figur 186 - Browsertest - Win 7 - Firefox 20.....	164
Figur 187 - Browsertest - Win 7 - Chrome 26.....	165
Figur 188 - Browsertest - Win 7 - Safari 5.1.....	166
Figur 189 - Browsertest - Win 8 - Explorer 10 .....	167
Figur 190 - Browsertest - Win 8 - Firefox 20.....	168
Figur 191 - Browsertest - Win 8 - Chrome 26.....	169
Figur 192 - Browsertest - Win 8 - Safari 5.1.....	170
Figur 193 - Browsertest - OSX Lion - Firefox 20.....	171
Figur 194 - Browsertest - OSX Lion - Chrome 26 .....	172
Figur 195 - Browsertest - OSX Lion - Safari 6.....	173
Figur 196 - Browsertest - IOS 6 - iPad3 Safari.....	174
Figur 197 - Browsertest - IOS 6 - iPhone 5 Safari.....	175
Figur 198 - Browsertest - Android 4.1- Samsung Galaxy SIII .....	176
Figur 199 - Browsertest - Android 4.1 - Samsung Galaxy Note 10.1 .....	177
Figur 200 - Screenshot - Gammel webside.....	178

## 17. Tabelbetegnelse

Tabel 1 - Database Symbolbetydning .....	19
Tabel 2- Javascript funktioner .....	42
Tabel 3 - Input validering - Opret Person .....	46
Tabel 4 - Input Validering - Opret Suppleant.....	47
Tabel 5 - Input Validering - Opret Repræsentant.....	47
Tabel 6 - Input Validering - Tilføj Til Bestyrelsen .....	47
Tabel 7 - Input Validering - Opdater Person.....	48
Tabel 8 - Input Validering - Opret Admin .....	49
Tabel 9 - Input Validering - Rediger Admin .....	50
Tabel 10 - Database Forespørgsler – Opret Person.....	51
Tabel 11 - Database Forespørgsler - Opret Admin .....	51
Tabel 12 - Database Forespørgsler - Opret Suppleant .....	52
Tabel 13 - Database Forespørgsler - Opret Repræsentant.....	52
Tabel 14 - Database Forespørgsler - Tilføj Til Bestyrelsen.....	52
Tabel 15 - Database Forespørgsler - Tilføj Billede .....	53
Tabel 16 - Database Forespørgsler – Select .....	53
Tabel 17 - Database Forespørgsler - Slet Person.....	53
Tabel 18 - Database Forespørgsler - Select .....	54
Tabel 19 - Sammenhæng mellem tabel, form og knap for getmember.php .....	56
Tabel 20 - Sammenhæng mellem tabel, form og knap for getiming.php .....	56
Tabel 21 - Input Validering - Metodebeskrivelse .....	58
Tabel 22 - Input Validering - Anvendte Metoder - Opret Person.....	58
Tabel 23 - Input Validering - Anvendte Metoder - Opret Medlemskab .....	59
Tabel 24 - Input Validering - Anvendte Metoder - Opdater Person.....	59
Tabel 25 - Input Validering - Anvendte Metoder - Slet Person .....	59
Tabel 26 - Input Validering - Anvendt Metoder - Opret Admin .....	59
Tabel 27 - Input Validering - Anvendte Metoder – Slet Admin .....	60
Tabel 28 - Input Validering - Anvendte Metoder - Rediger Admin.....	60
Tabel 29 - SQL Injection funktioner .....	60
Tabel 30 - HTML Valideringstests .....	66
Tabel 31 - CSS Valideringstests.....	68
Tabel 32 – Browsertests .....	69
Tabel 33 - Person test data 1.....	70
Tabel 34 - Person test data 2.....	71
Tabel 35 - Person test data 3.....	71
Tabel 36 - Sommerhus test data 1.....	71
Tabel 37 - Suppleant test data 1.....	71
Tabel 38 - Suppleant test data 2.....	71
Tabel 39 - Repræsentant test data 1.....	72
Tabel 40 - Repræsentant test data 2.....	72
Tabel 41 - Bestyrelsen test data 1 .....	72

Tabel 42 - Bestyrelsen test data 2 .....	72
Tabel 43 - Billede test data 1 .....	72
Tabel 44 - Blackbox test uden forventede fejl.....	75
Tabel 45 - Blackbox test med forventede fejl.....	77
Tabel 46 - Admin test data 1 .....	77
Tabel 47 - Admin test data 2 .....	77
Tabel 48 - Blackbox test uden forventede fejl.....	78
Tabel 49 - Blackbox test med forventede fejl.....	78
Tabel 50 - Blackbox test – Login .....	79



# 18. Litteraturliste

## 18.1. Bøger

**Connolly, Thomas M. & Begg, Carolyn E. (2010)**

Database Systems, A practical approach to design, implementation, and management

5.udgave

Pearson Education Inc.

**Bruegge, Bernd & Dutoit, Allen H. (2004)**

Object-Oriented Software Engineering, Using UMK, Patterns and Java

2.udgave

Pearson Education Inc.

**Meloni, Julie C. (2005)**

PHP, MySQL and Apache

2.udgave

Sams publishing

**Hayes, Deidre (2007)**

HTML

1.udgave, 1.oplag

Forlaget Libris

**Nixon, Robin (2009)**

Learning PHP, MySQL, and Javascript

1.udgave

O'reilly media

## 18.2. Internet

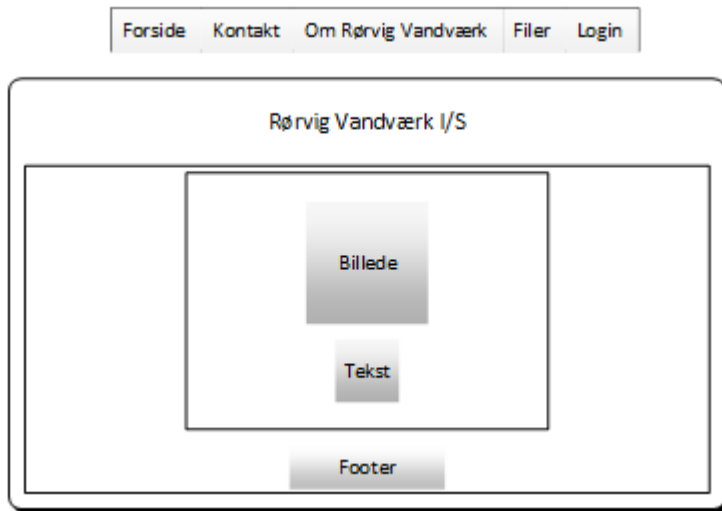
<http://www.w3schools.com>

<http://www.php.net>

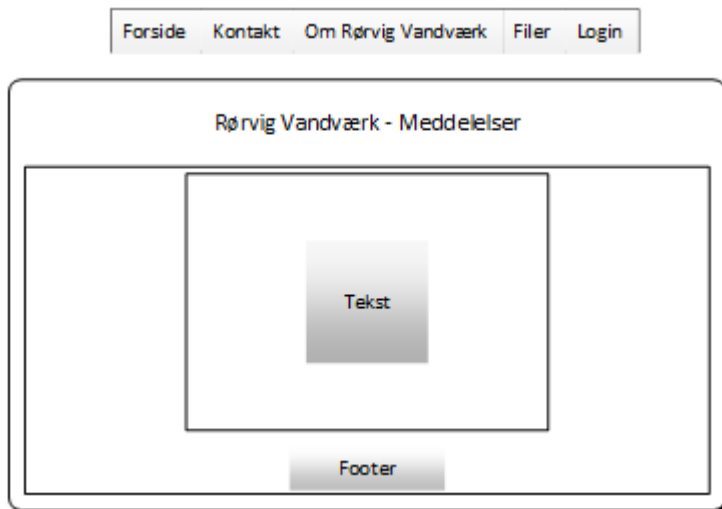
<http://en.wikipedia.org>

# 19. Bilag

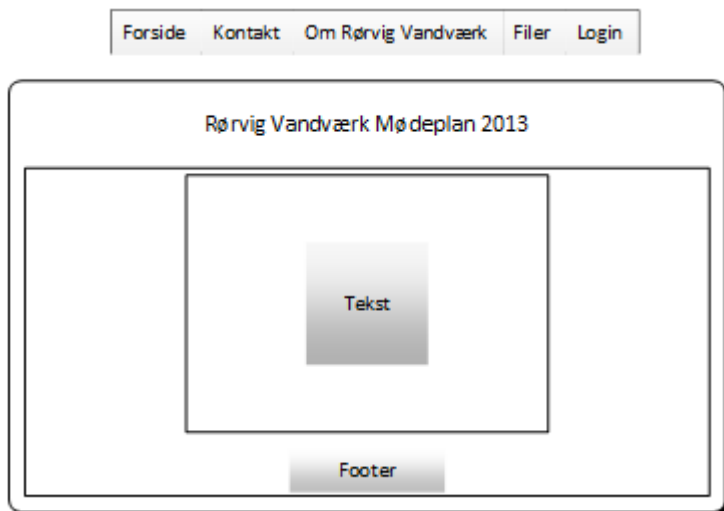
## 19.1. Wireframes



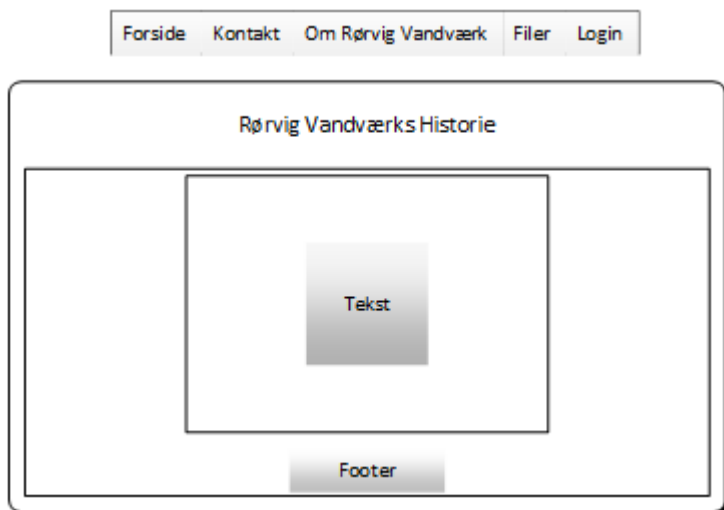
Figur 32 - Wireframe – Index



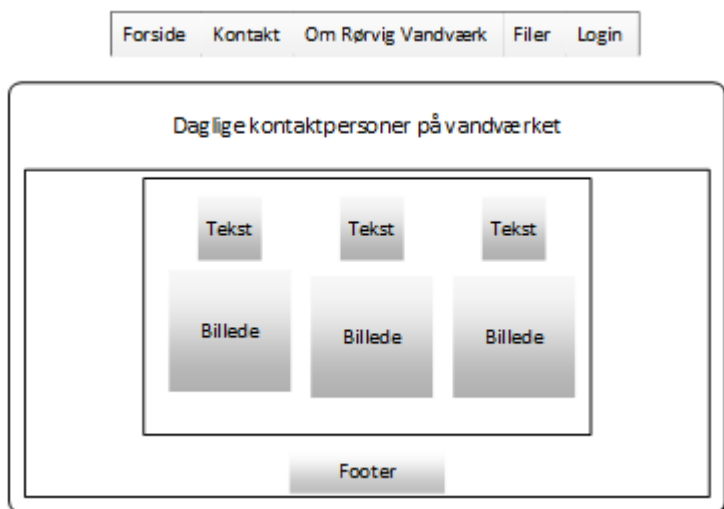
Figur 33 - Wireframe – Meddelelser



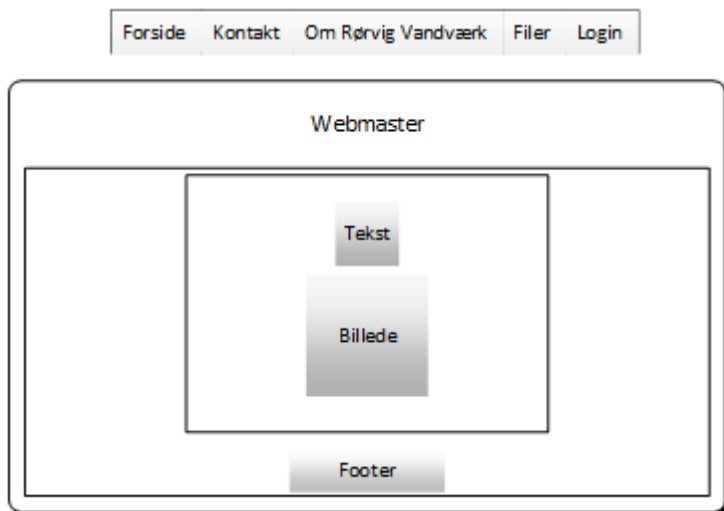
Figur 34 - Wireframe – Mødeplan



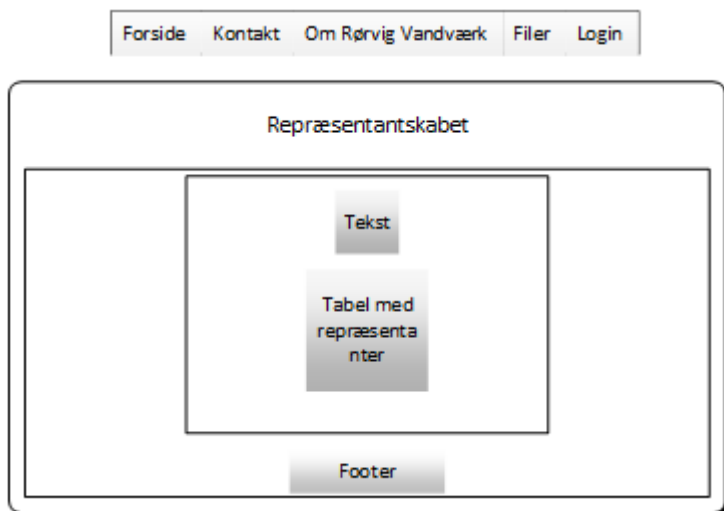
Figur 35 - Wireframe – Historie



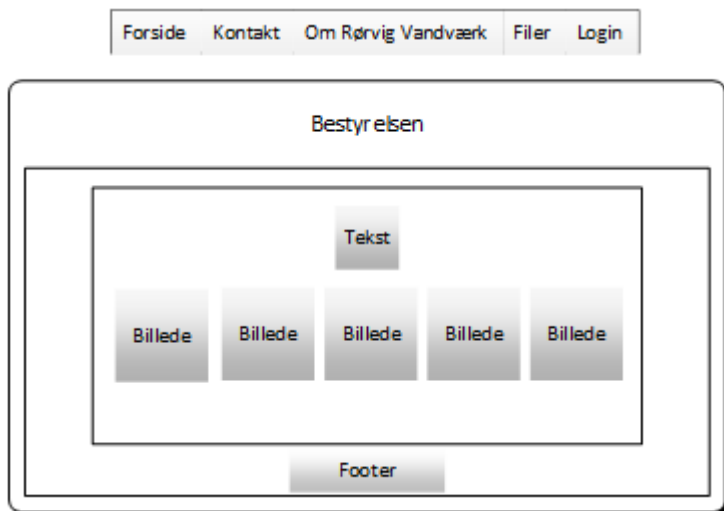
Figur 36 - Wireframe – Kontaktpersoner



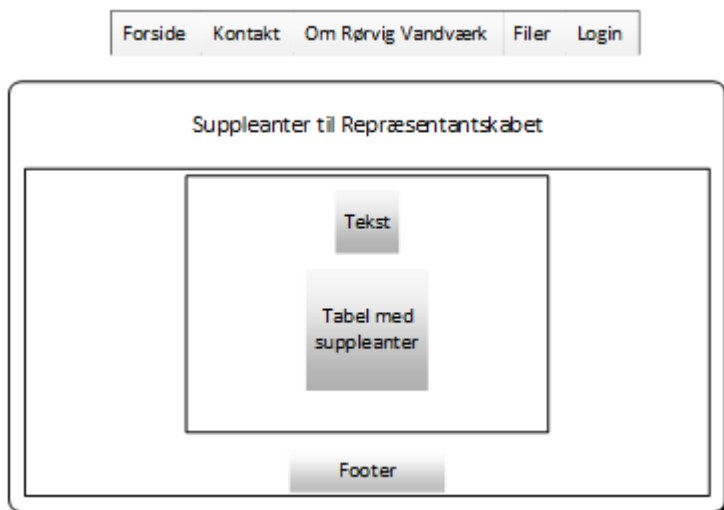
Figur 37 - Wireframe – Webmaster



Figur 38 - Wireframe – Medlemmer



Figur 39 - Wireframe – Bestyrelsen



Figur 40 - Wireframe – Suppleanter

### Person Administration

PersonSuppleantRepræsen  
tantBestyrelse  
nSommerh  
usBilleder

Suppleant

Repræsentant

Navn
Adresse
Postnummer
Tlf
Mobil
E-mail
R-siden
Sommerhus adr.
Sommerhus postnr.
Sommerhus tlf.

Tilføj

Vælg filUpload

Vælg person  Slet

Vælg person

Suppleant: Skriv valg år	Suppleant: Skriv prioritet	Tilføj
Repræsentant: Skriv valg år	Repræsentant: Skriv sup_t_b	Tilføj
Bestyrelsen: Skriv valg	Vælg titel <input type="checkbox"/>	Tilføj

Vælg person

navn	adr	postnr	tlf	mobil	E-mail	R-siden
Sommerhus adr		Sommerhus postnr	Sommerhus tlf			
Repræsentant valg		Repræsentant sup_t_B				
Suppleant valg år		Suppleant prioritet				
Bestyrelse valg		Bestyrelse titel <input type="checkbox"/>				

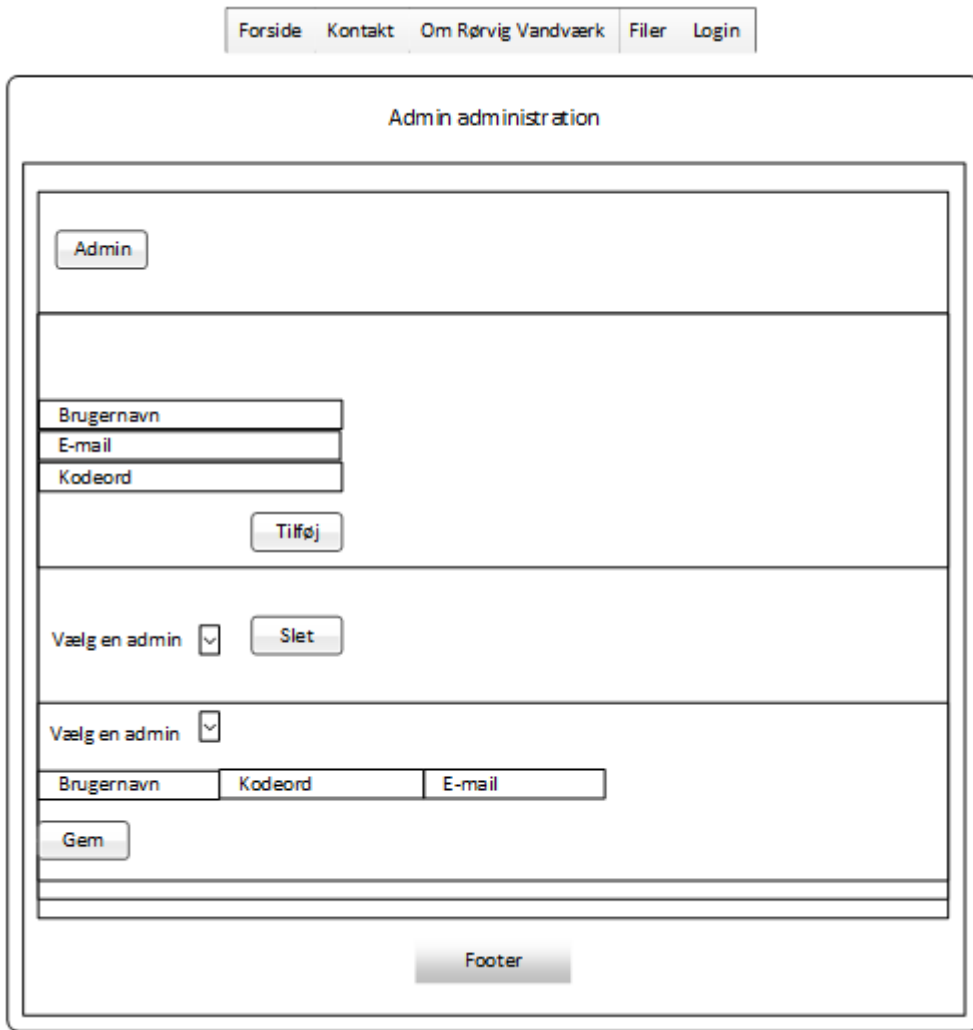
Gem

Person billeder

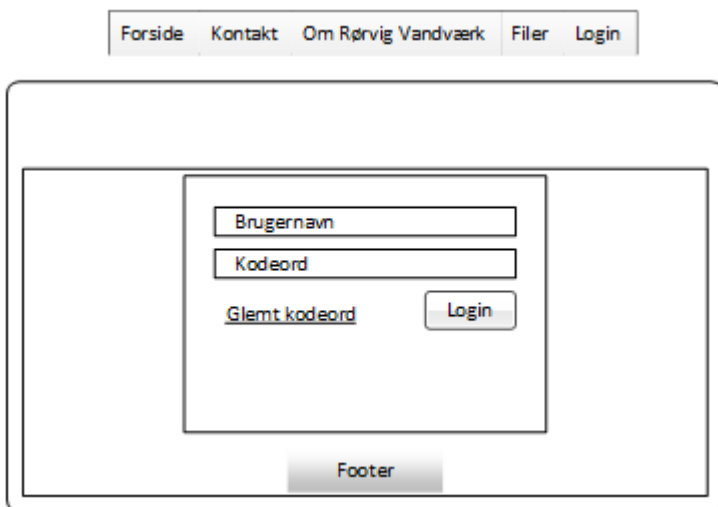
Vælg filUpload

Footer

Figur 41 - Wireframe - Person Administration

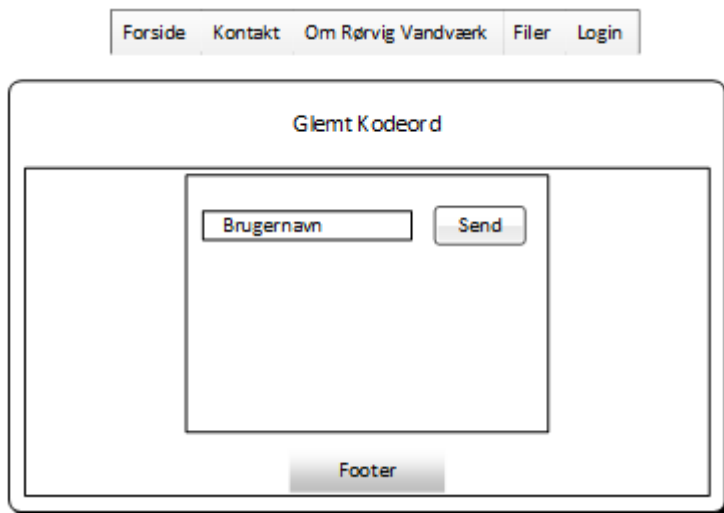


Figur 42 - Wireframe - Admin Administration



Figur 43 - Wireframe – Login





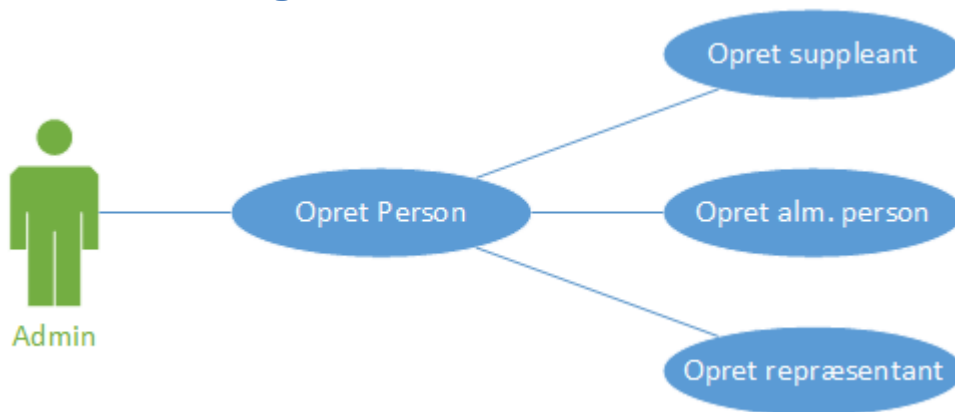
Figur 44 - Wireframe - Glemt Login

## 19.2. Comps



Figur 45 - Comps - Baggrundsbillede

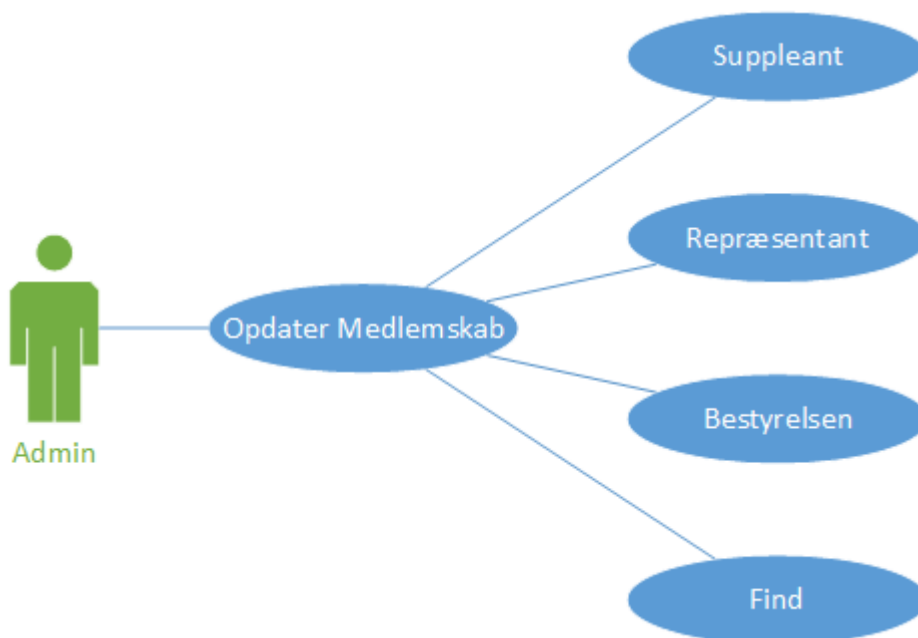
### 19.3. Use case diagrammer



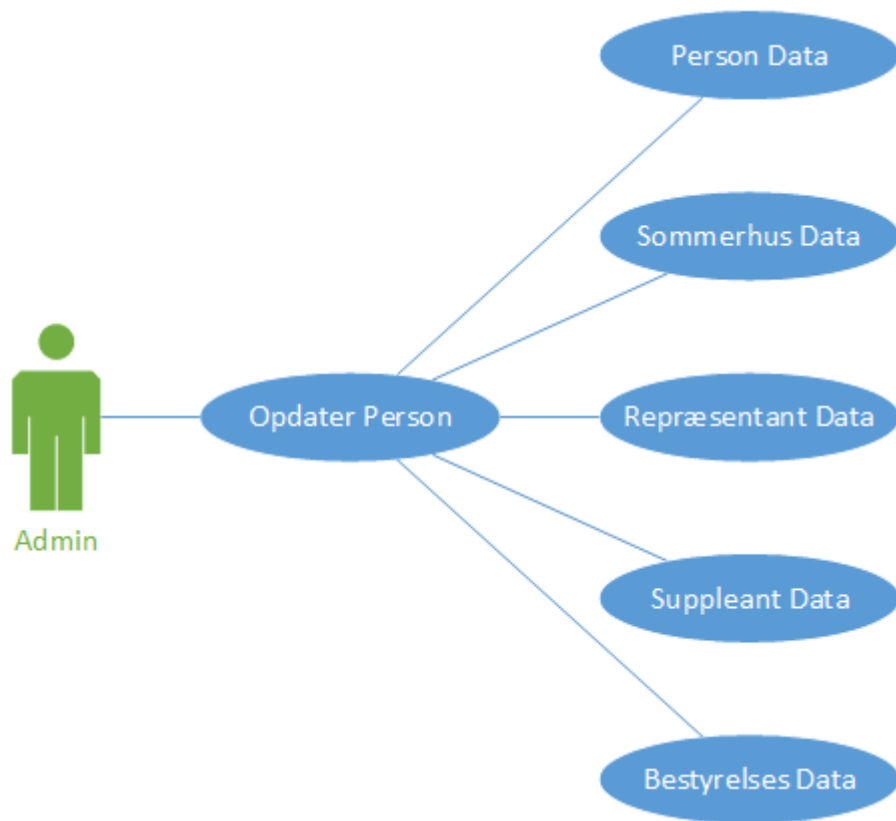
Figur 46 - Use Case - Opret person



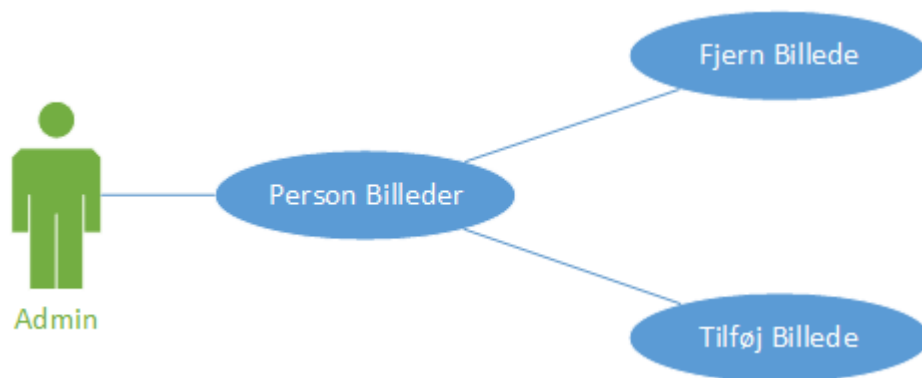
Figur 47 - Use Case - Slet person



Figur 48 - Use Case - Opdater Medlemskab

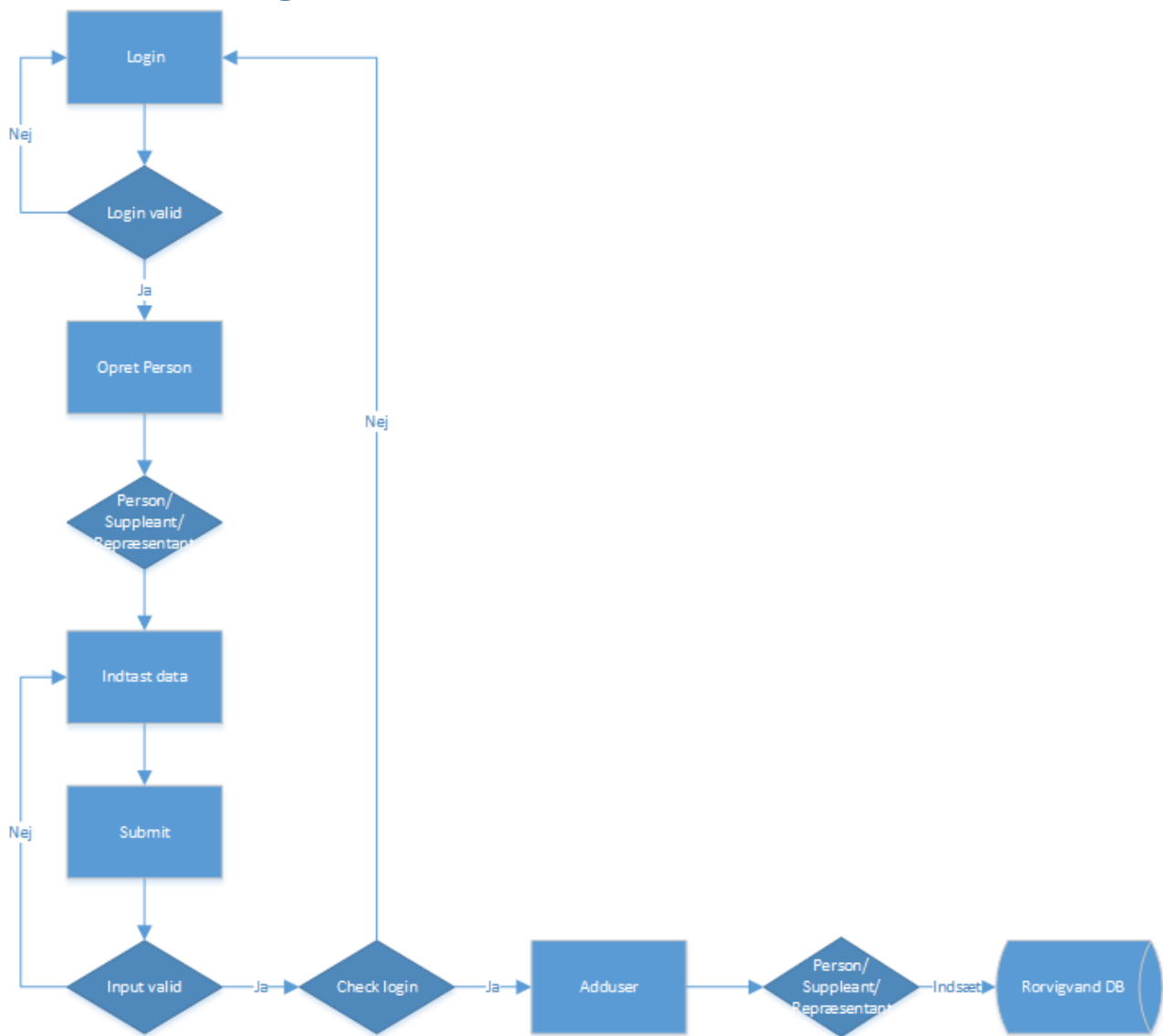


Figur 49 - Use Case - Opdater Person

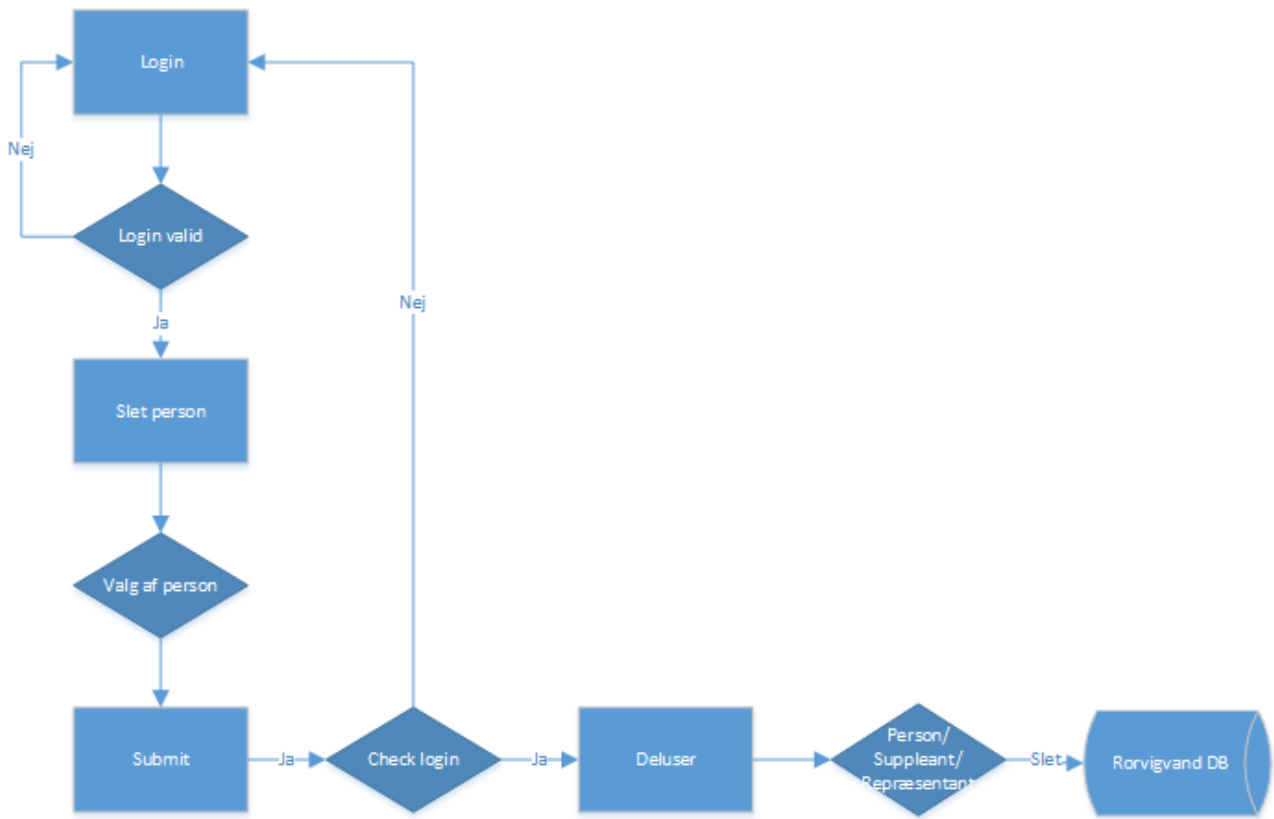


Figur 50 - Use Case - Person billeder

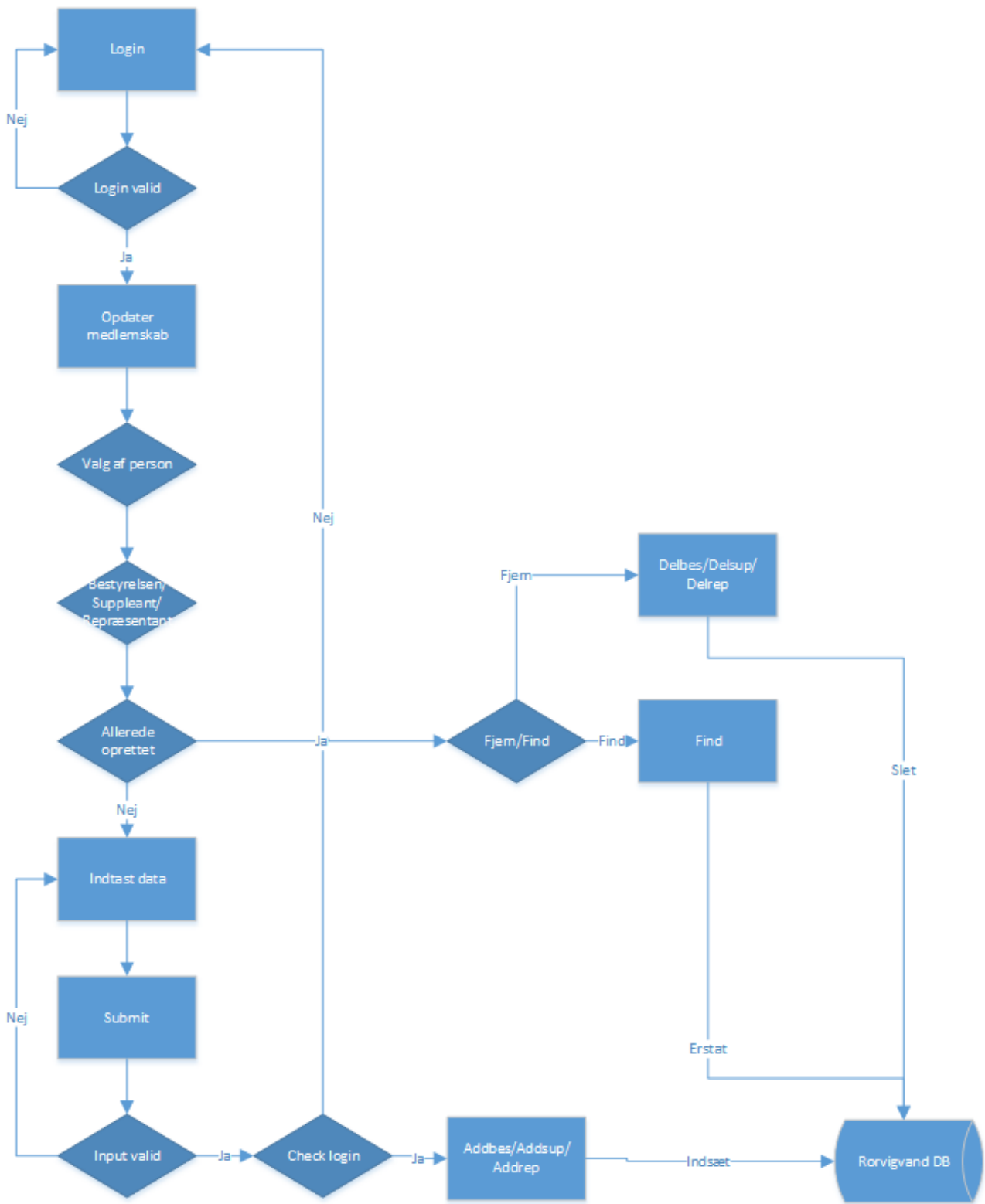
## 19.4. Flowchart diagrammer



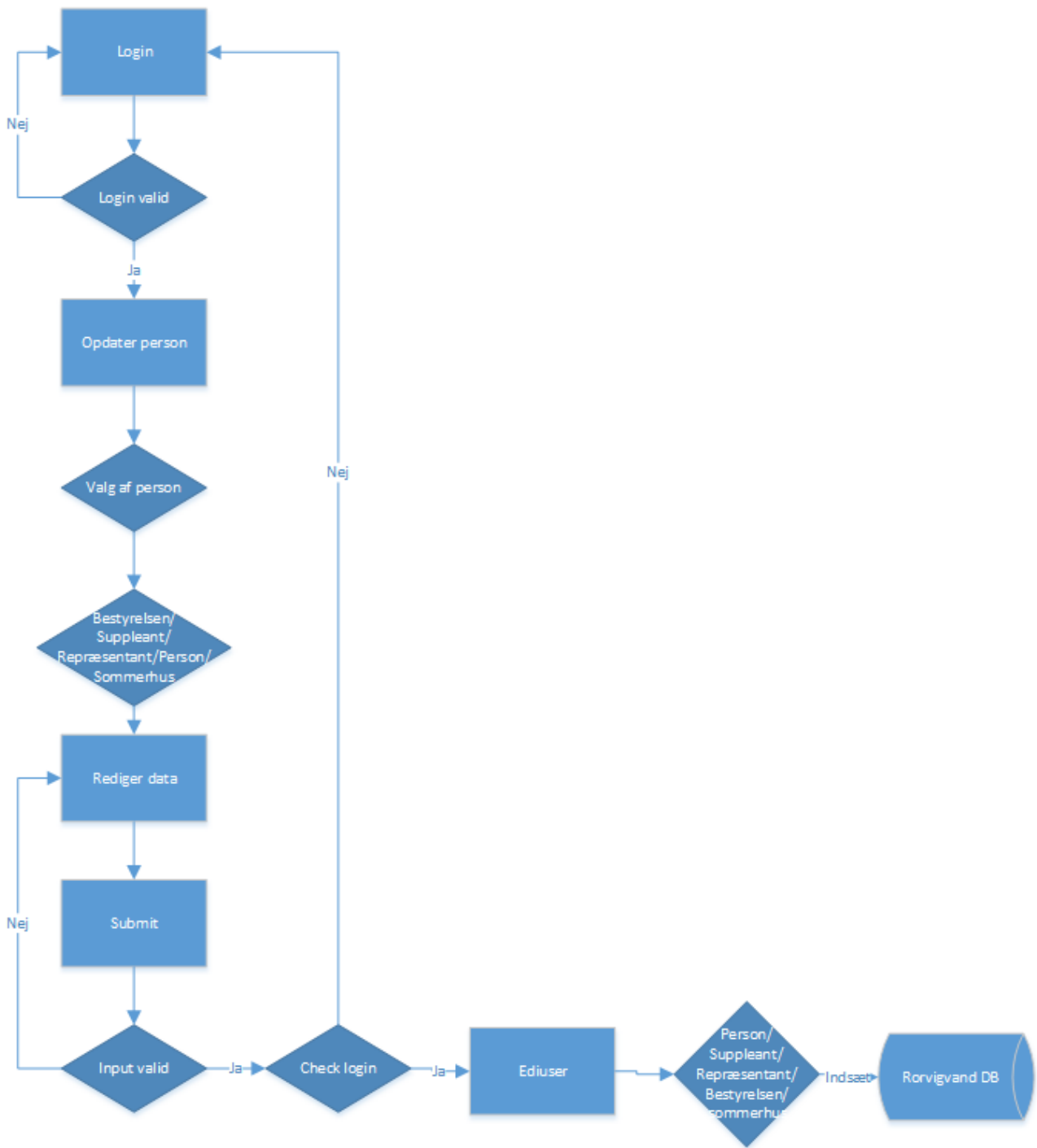
Figur 51 - Flowchart - Opret Person



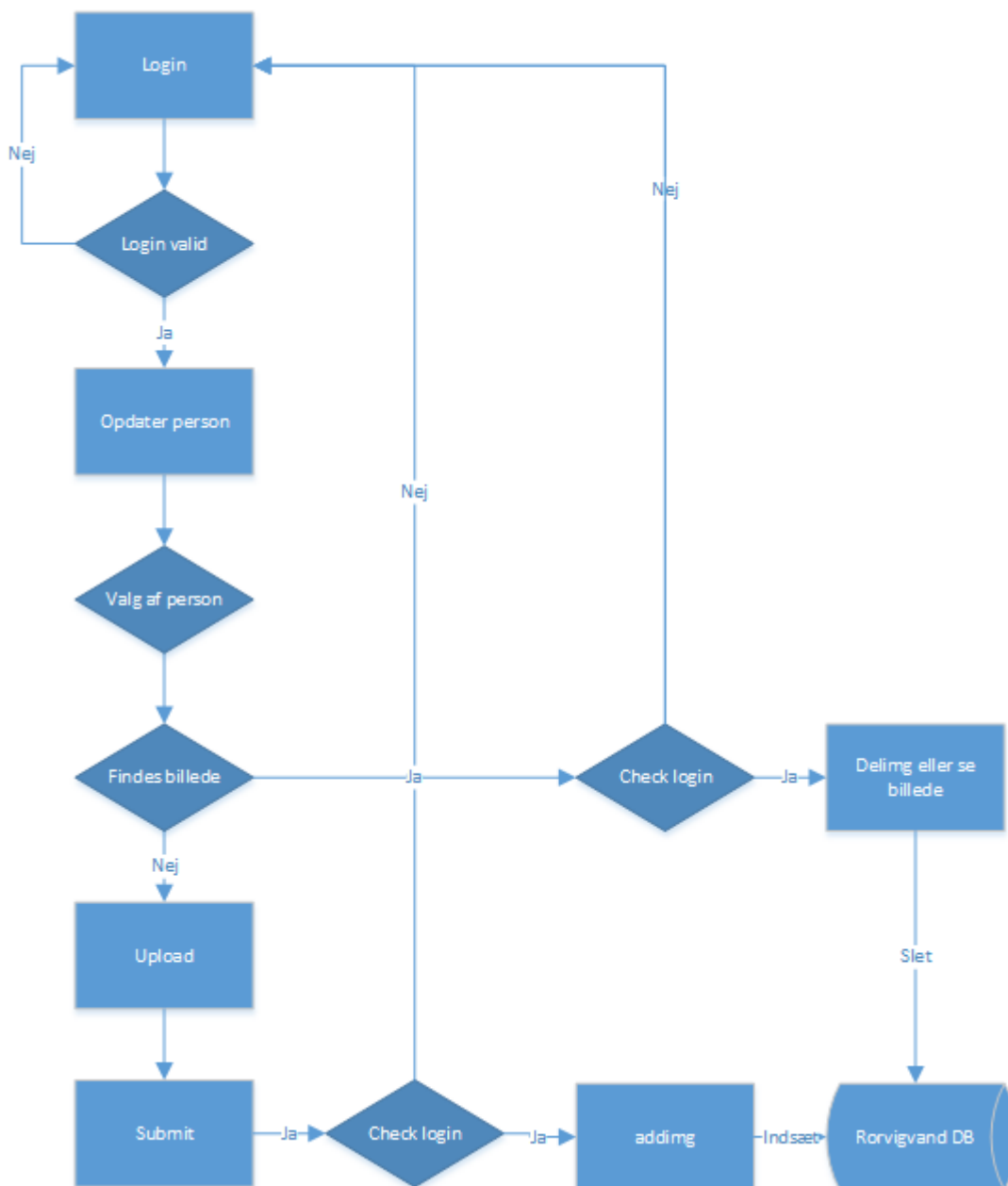
Figur 52 - Flowchart - Slet Person



Figur 53 - Flowchart - Opdater Medlemskab

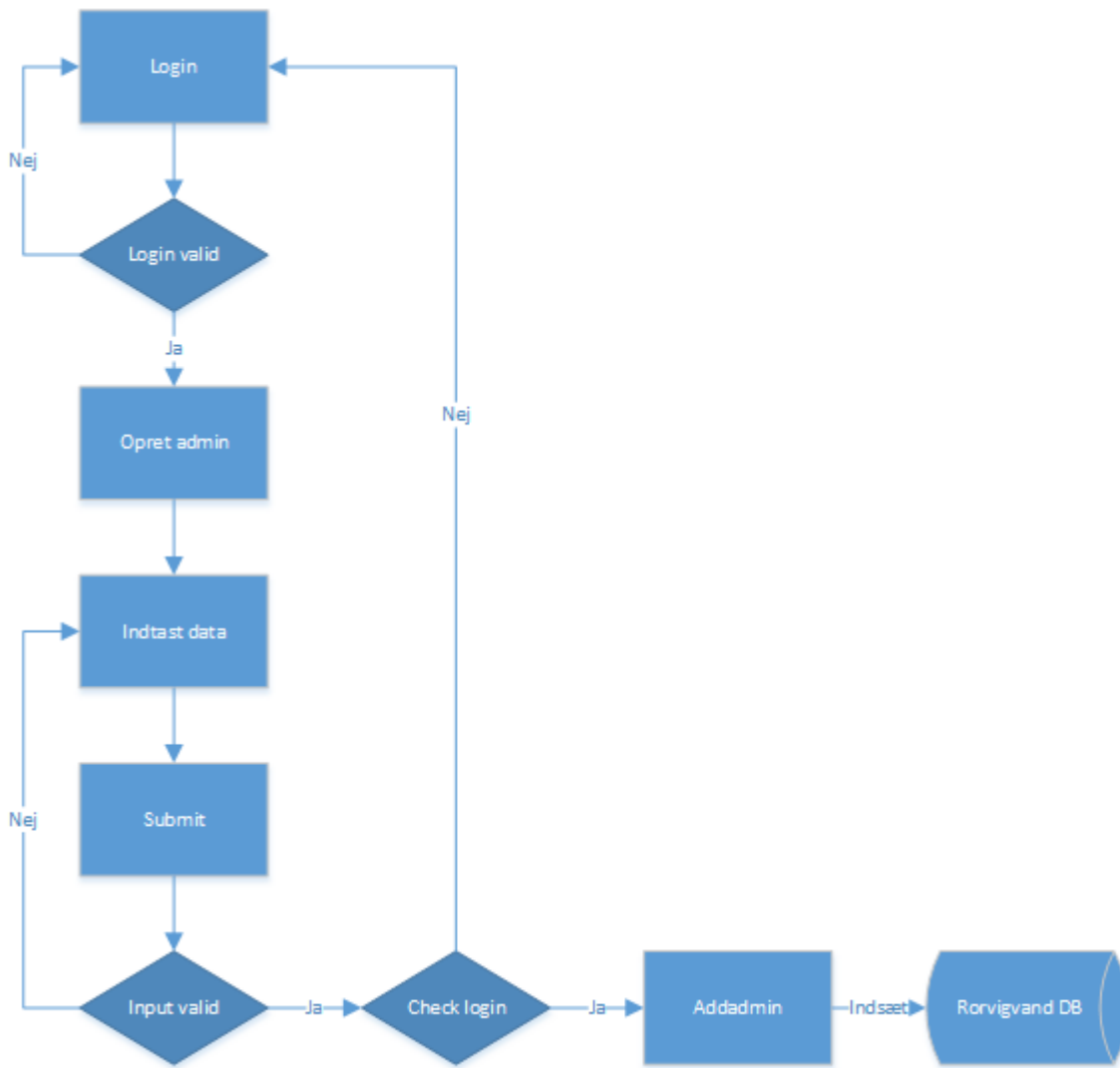


Figur 54 - Flowchart - Opdater Person

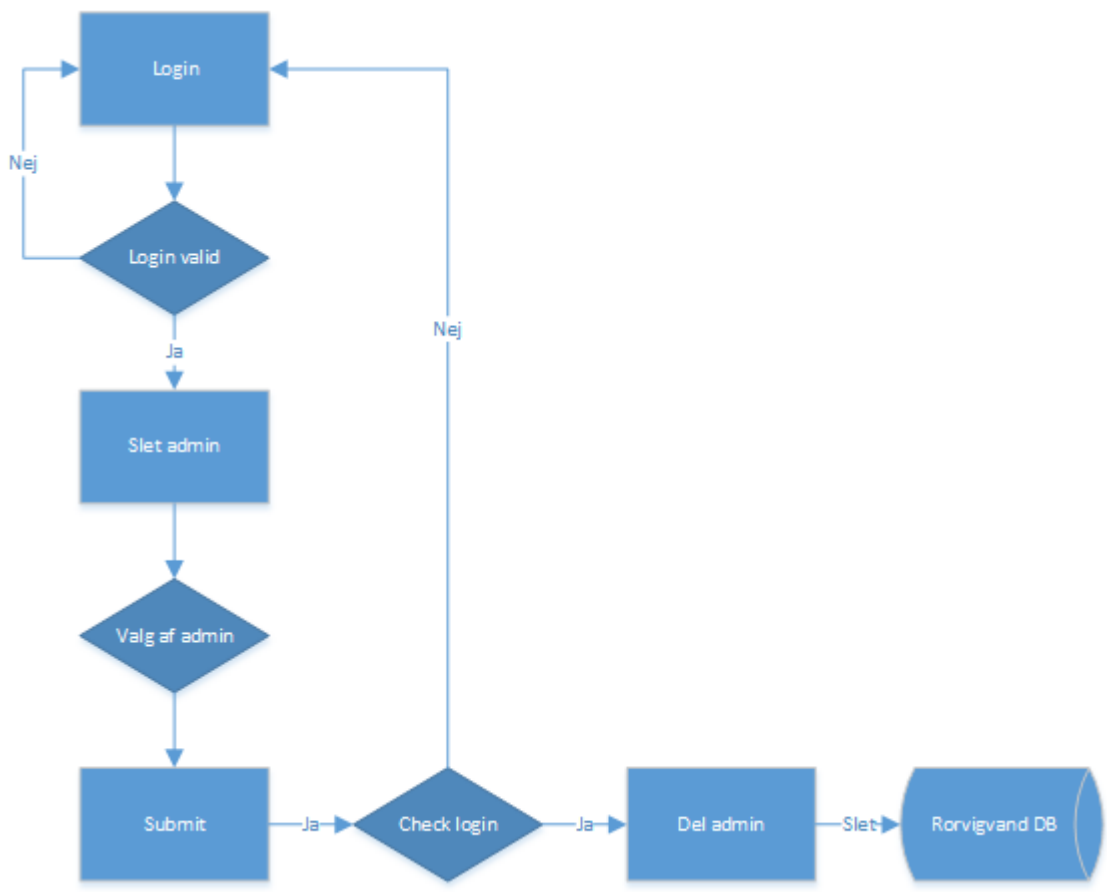


Figur 55 - Flowchart - Person Billeder

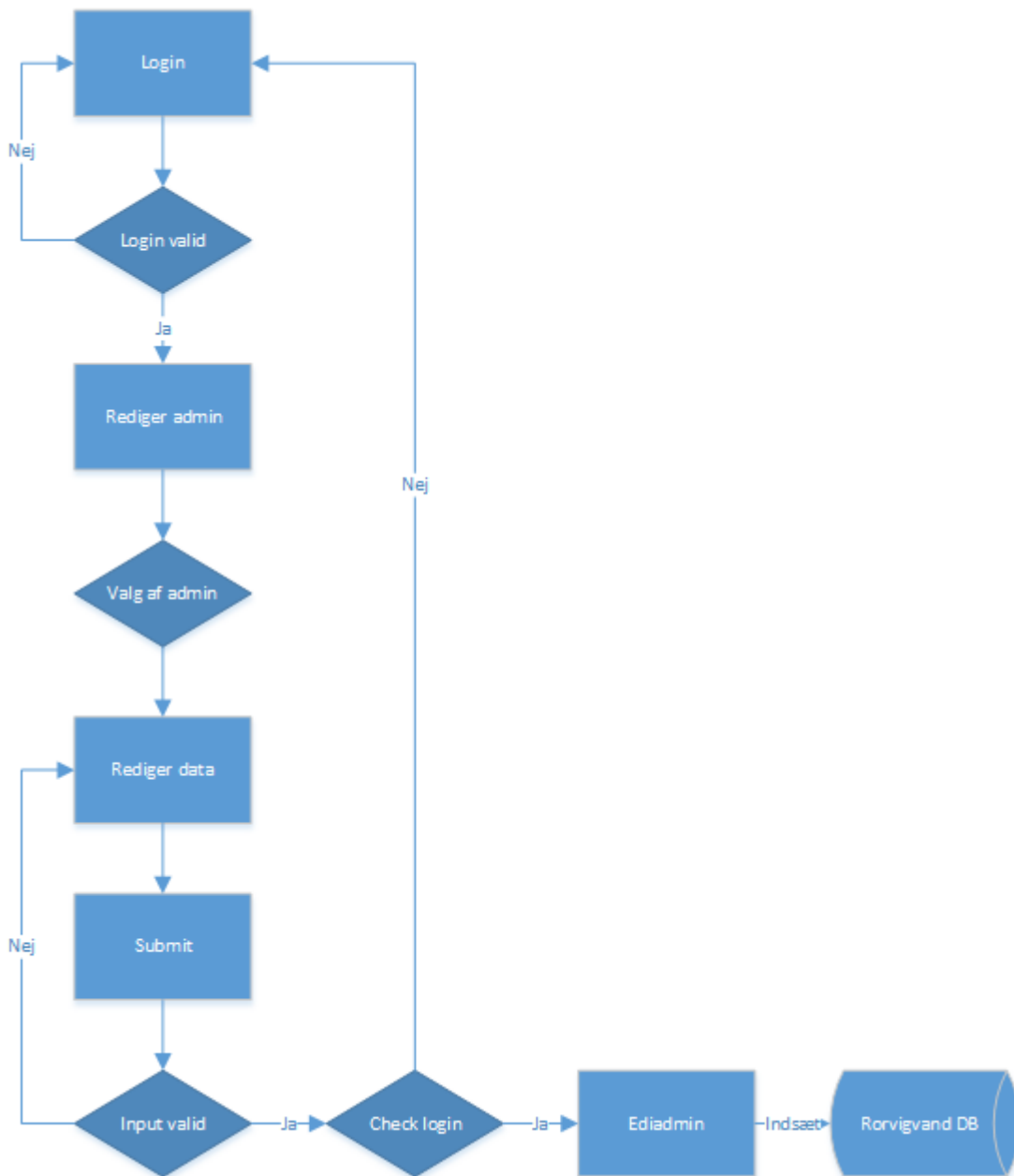




Figur 56 - Flowchart - Opret Admin

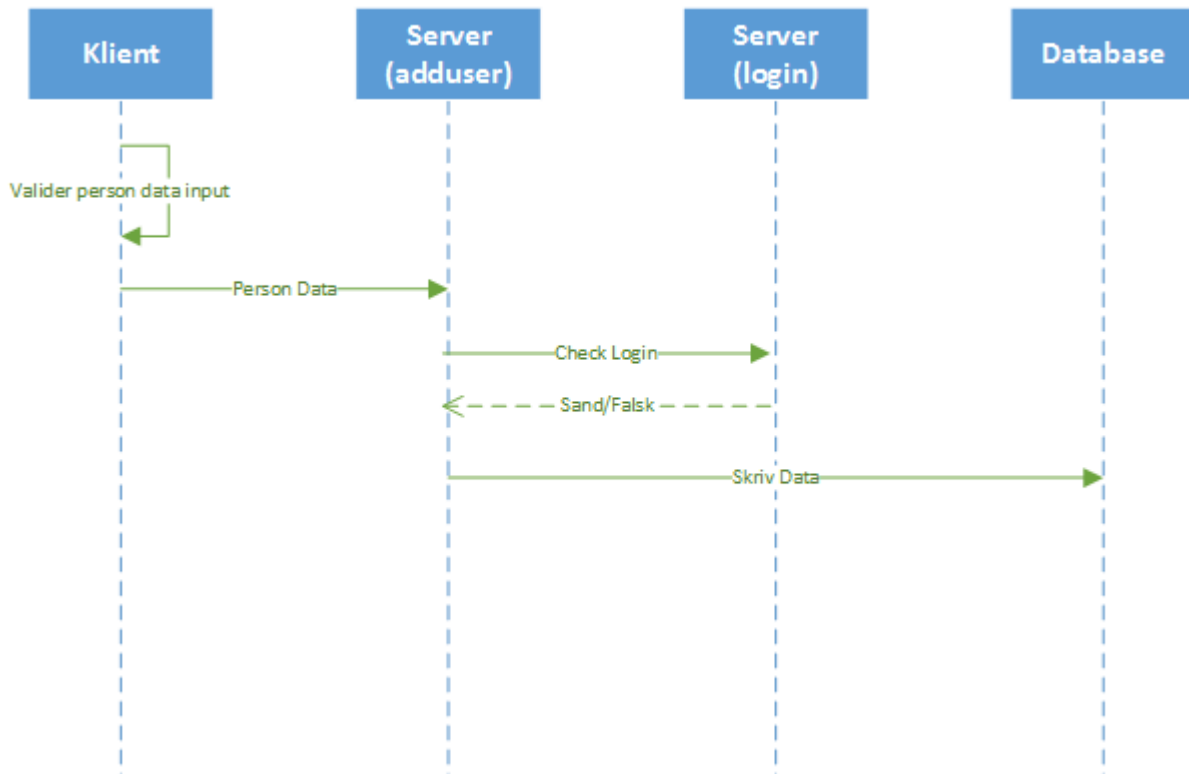


Figur 57 - Flowchart - Slet Admin

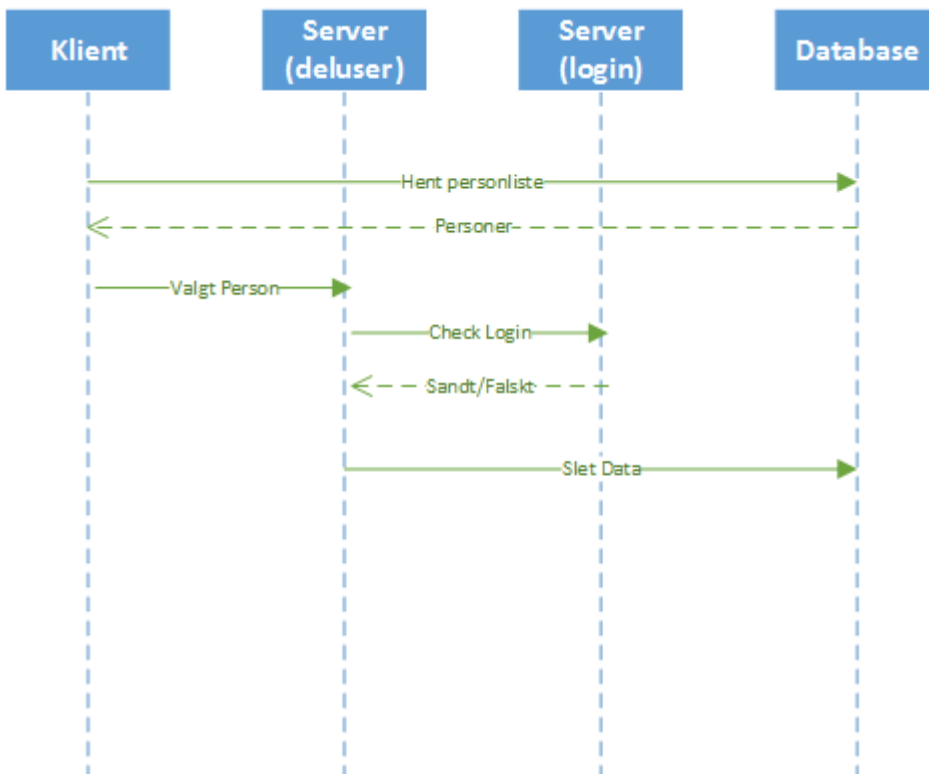


Figur 58 - Flowchart - Rediger Admin

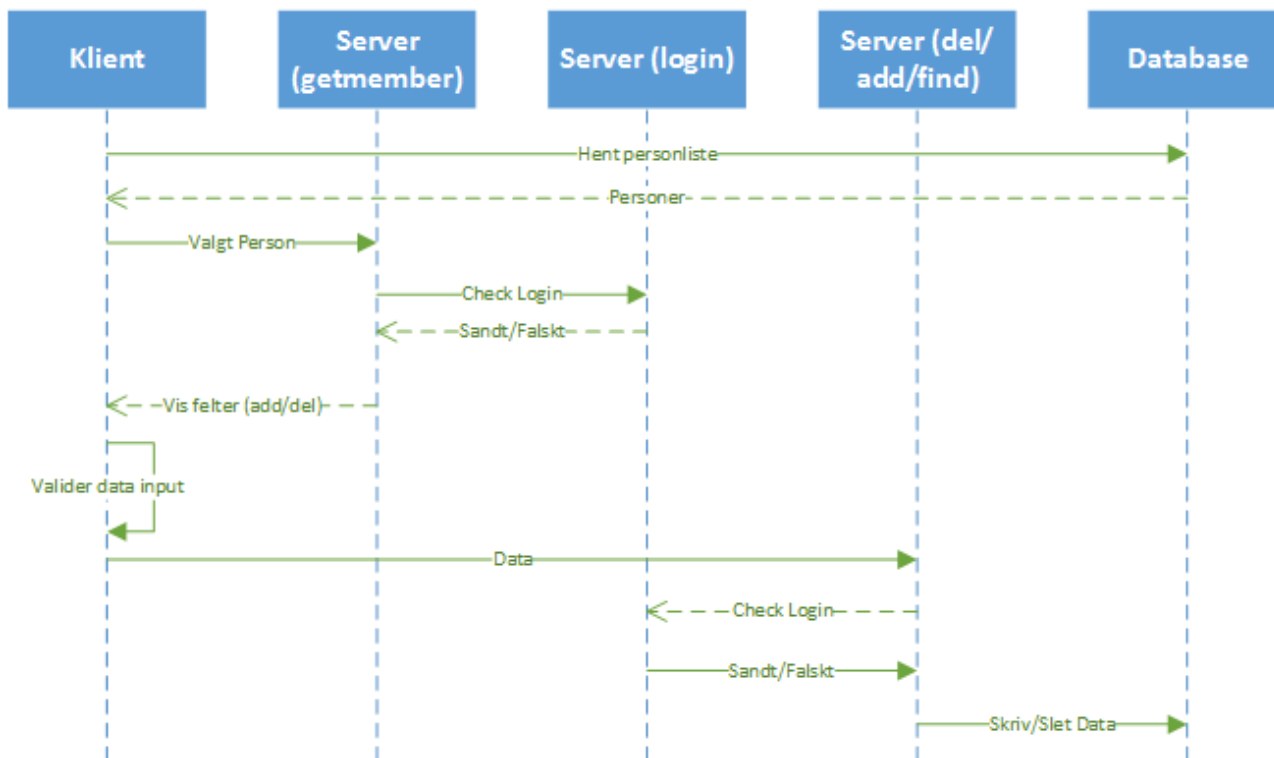
## 19.5. Sekvensdiagrammer



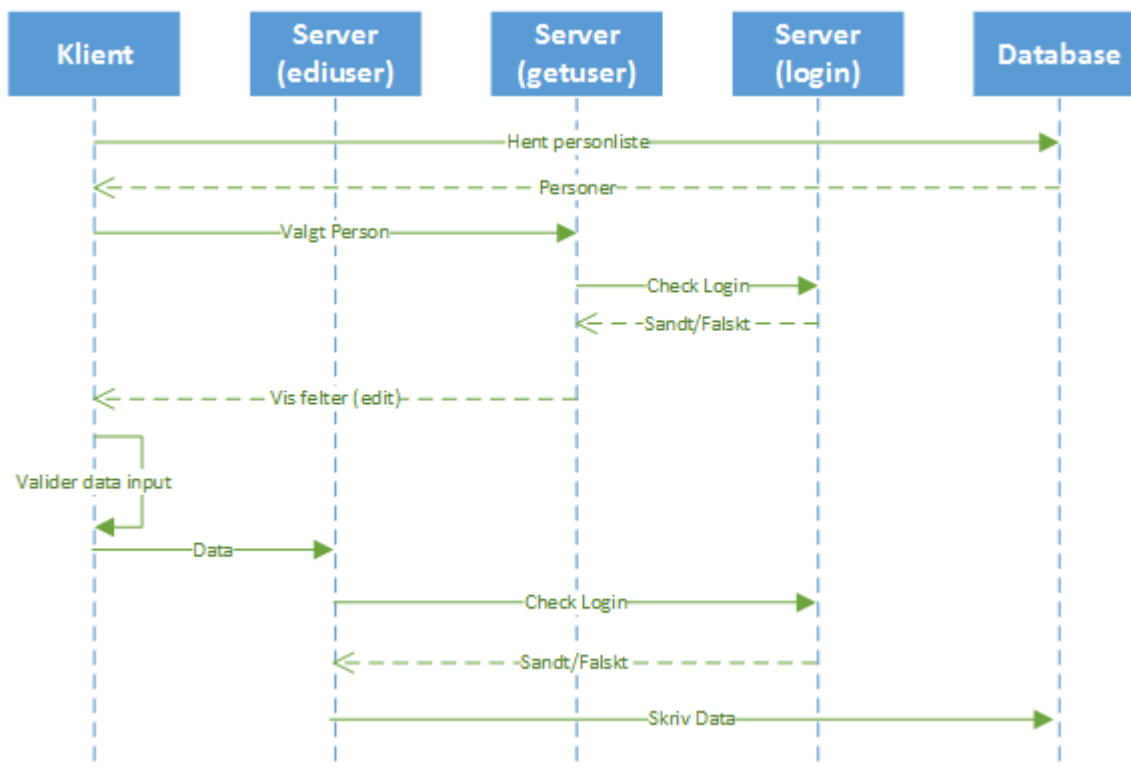
Figur 59 - Sekvensdiagram - Opret Person



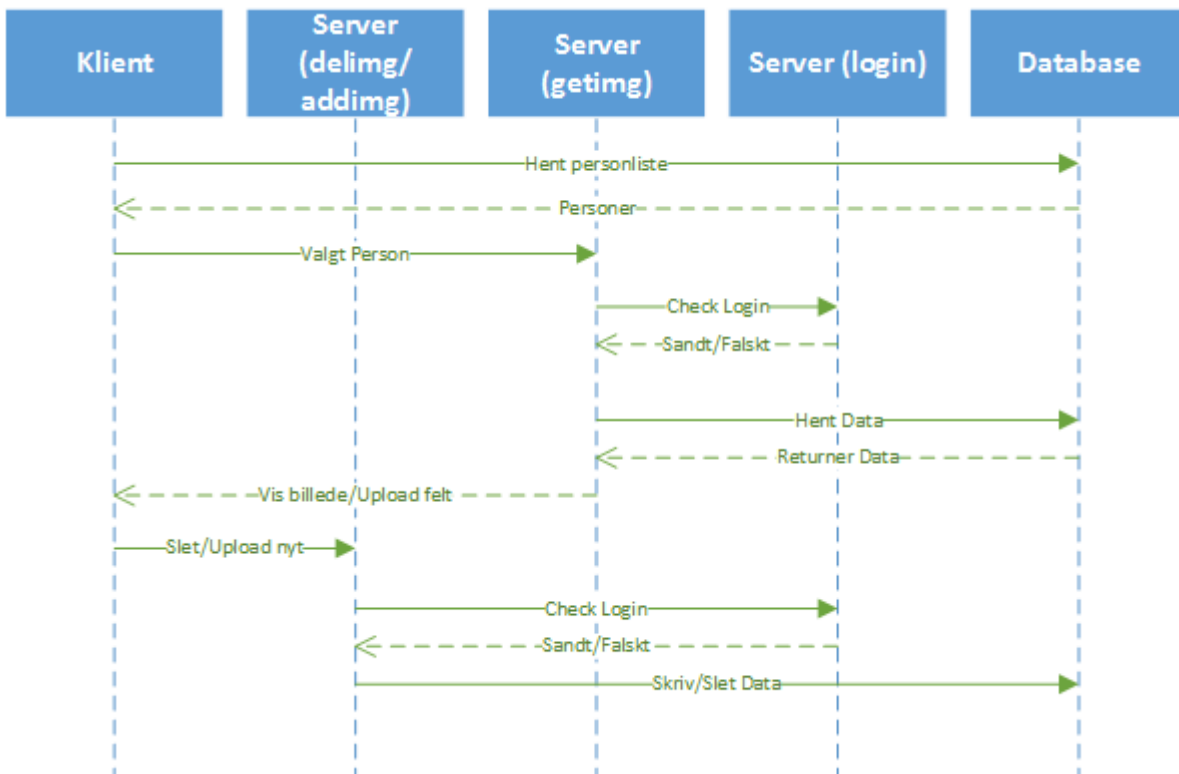
Figur 60 - Sekvensdiagram - Slet Person



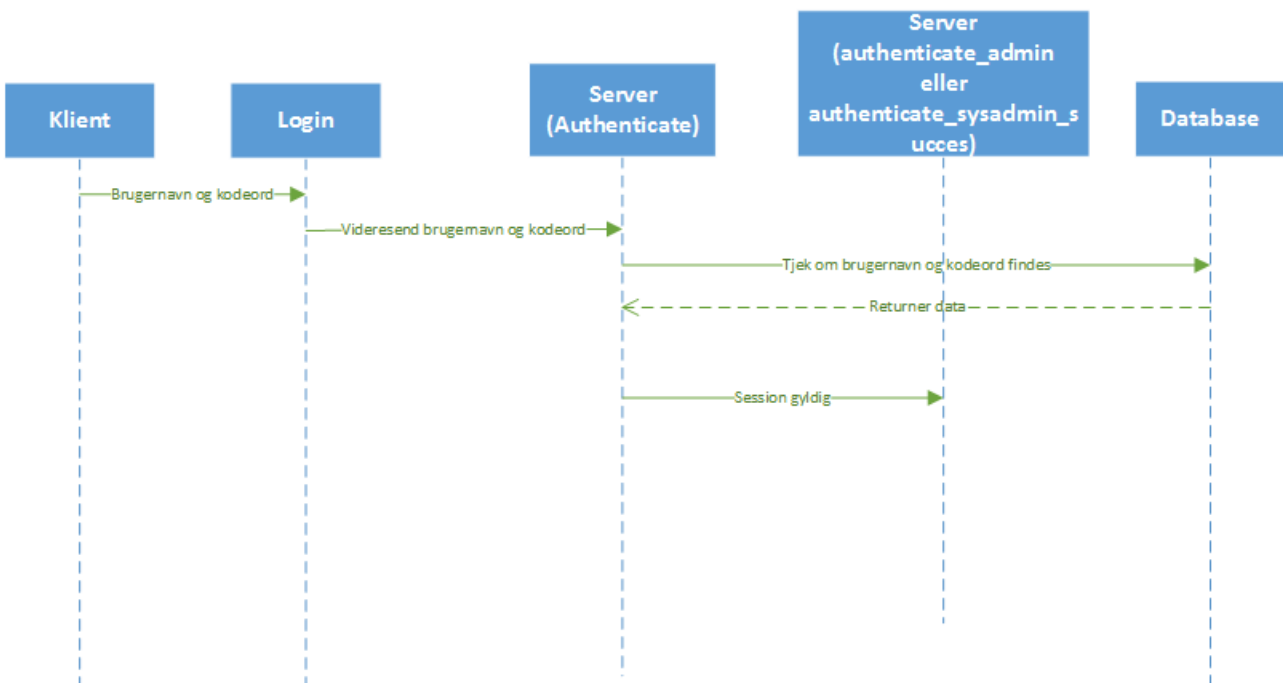
Figur 61 - Sekvensdiagram - Opdater Medlemskab



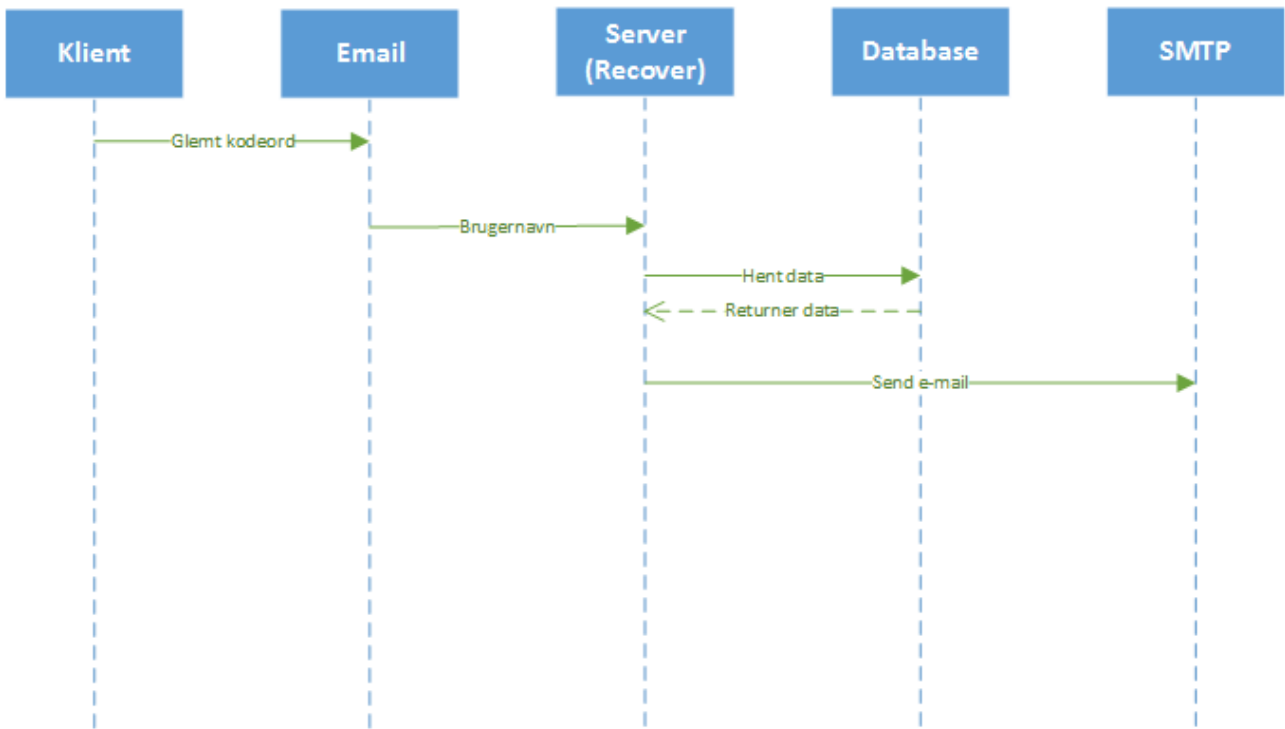
Figur 62 - Sekvensdiagram - Opdater Person



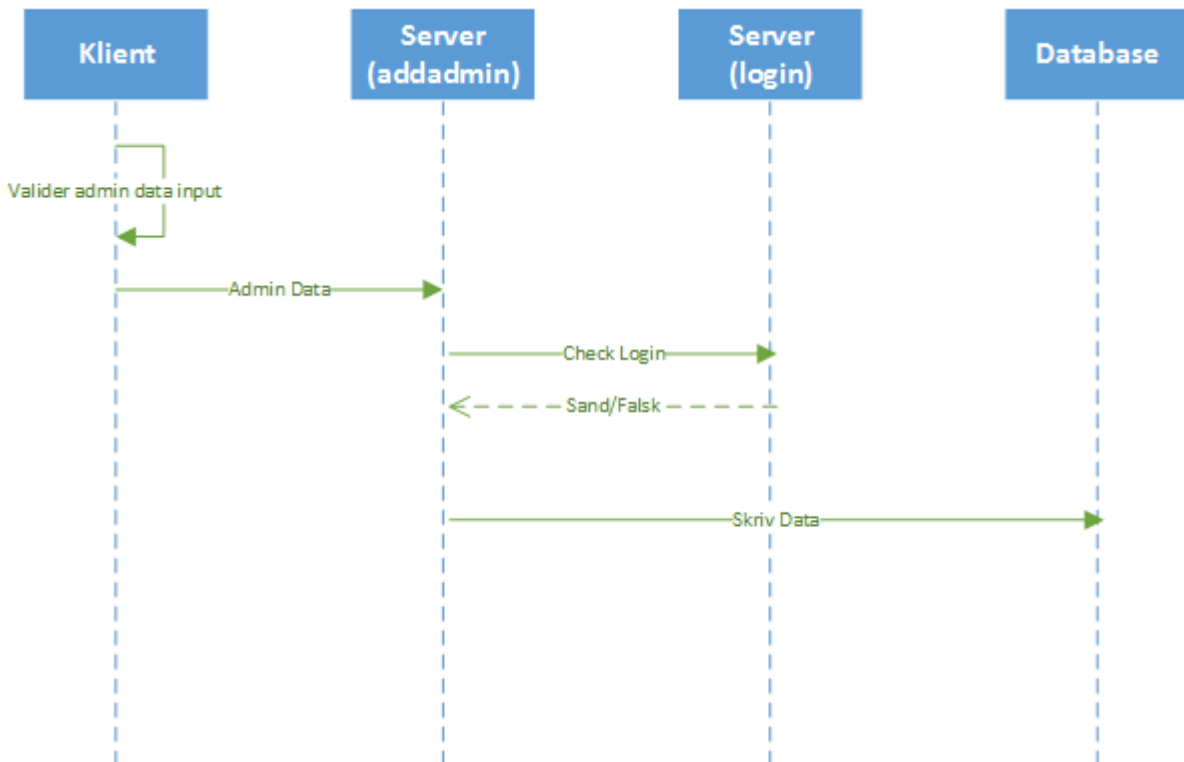
Figur 63 - Sekvensdiagram - Person Billeder



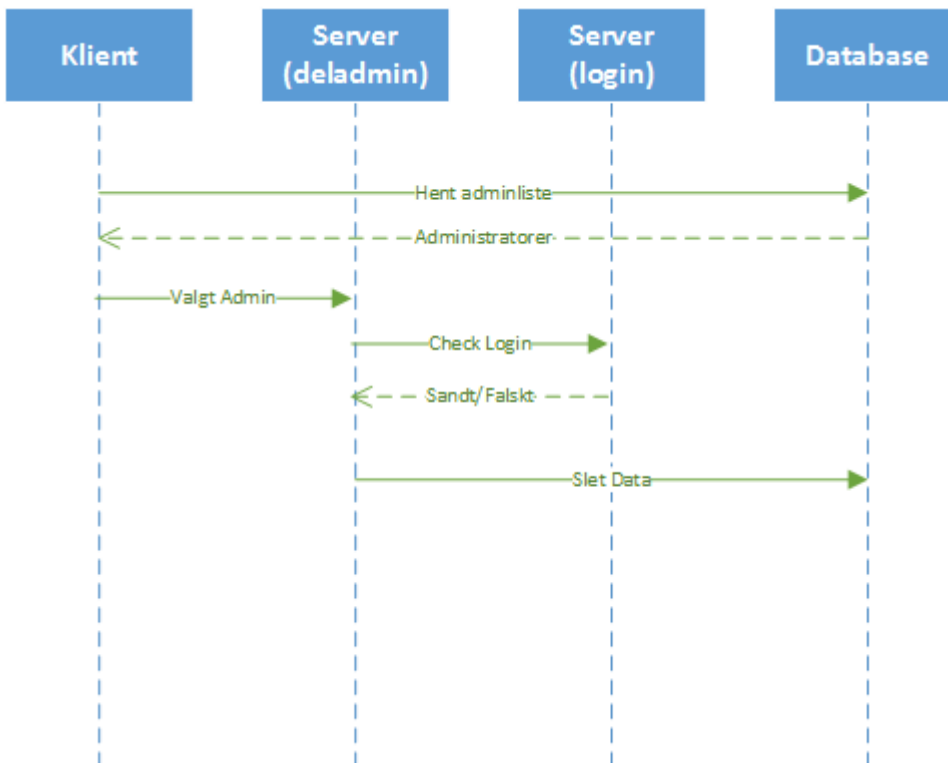
Figur 64 - Sekvensdiagram – Login



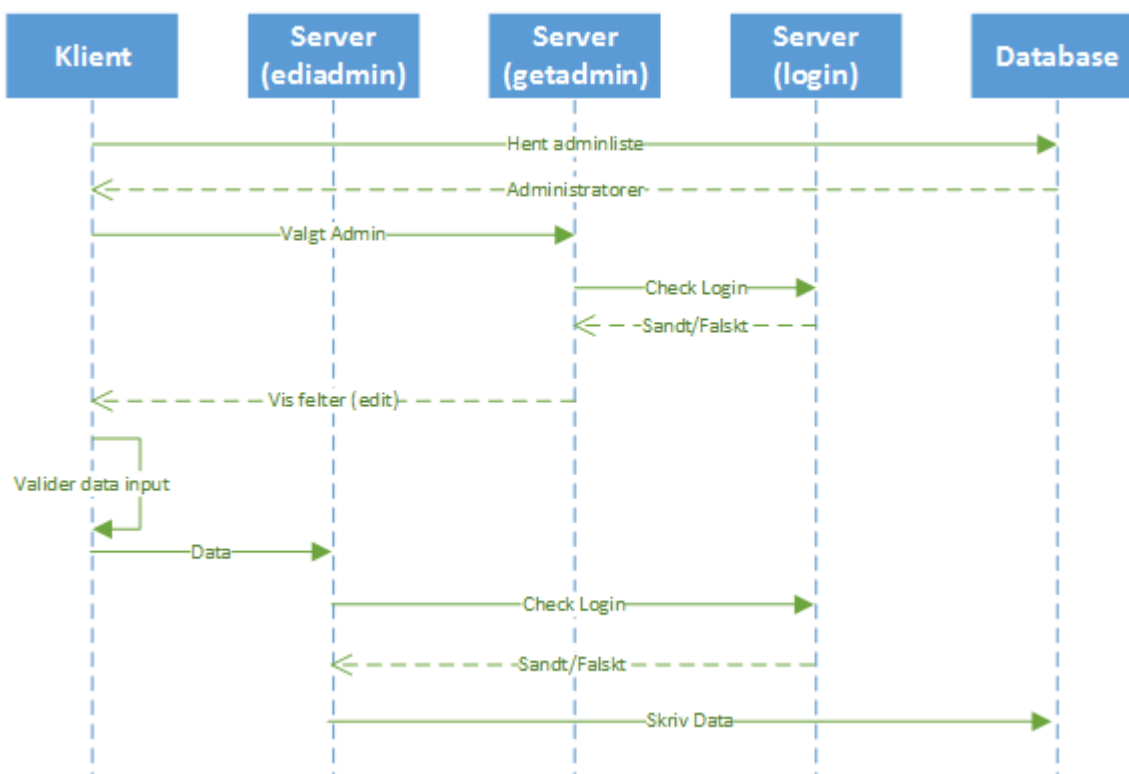
Figur 65 - Sekvensdiagram - Glemt Login



Figur 66 - Sekvensdiagram - Opret Admin



Figur 67 - Sekvensdiagram - Slet Admin



Figur 68 - Sekvensdiagram - Rediger Admin



## 19.6. Blackbox test

	pid	navn	adr	pnr	tlf	mobil	e_mail	r_siden
<input type="checkbox"/> Ret <input type="checkbox"/> Kopi <input type="checkbox"/> Slet	65	Test	testvej 1	2000	12345678	12345678	test@test.dk	

Figur 69 - Blackbox test - Opret alm. Person

	sid	pid	adr	postnr	tlf
--	-----	-----	-----	--------	-----

Figur 70 - Blackbox test – Person med fast adresse

	su_id	pid	sid	prioritet	valg_aar
<input type="checkbox"/> Ret <input type="checkbox"/> Kopi <input type="checkbox"/> Slet	143	65	NULL	1	2013

Figur 71 - Blackbox test - Opret suppleant med fast adresse

	su_id	pid	sid	prioritet	valg_aar
<input type="checkbox"/> Ret <input type="checkbox"/> Kopi <input type="checkbox"/> Slet	143	65	40	1	2013

Figur 72 - Blackbox test - Suppleant uden fast adresse

	rid	pid	sid	valg_aar	sup_t_B
<input type="checkbox"/> Ret <input type="checkbox"/> Kopi <input type="checkbox"/> Slet	153	65	NULL	2013	NULL

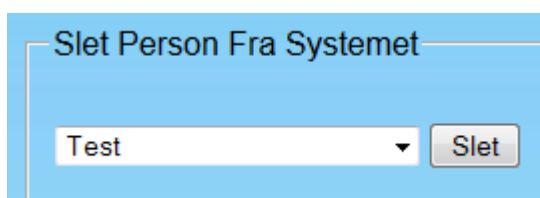
Figur 73 - Blackbox test – Opret repræsentant med fast adresse

	rid	pid	sid	valg_aar	sup_t_B
<input type="checkbox"/> Ret <input type="checkbox"/> Kopi <input type="checkbox"/> Slet	153	65	41	2013	NULL

Figur 74 - Blackbox test - Repræsentant uden fast adresse

	mid	pid	name	type	size	img
<input type="checkbox"/> Ret <input type="checkbox"/> Kopi <input type="checkbox"/> Slet	184	65	Test.jpg	image/jpeg	6772	[BLOB - 6,6 KiB]

Figur 75 - Blackbox test - Tilføj billede til person



Figur 76 - Blackbox test - Slet person

Opdater Medlemskab

Test

**Suppleant**

Medlem: Nej 2013 1 Tilføj

Figur 77 - Blackbox test - Opdater medlemskab - Tilføj suppleant

		su_id	pid	sid	prioritet	valg_aar
<input type="checkbox"/>	Ret	144	65	41	1	2013

Figur 78 - Blackbox test - Opdater medlemskab - Tilføj suppleant DB

Opdater Medlemskab

Test

**Suppleant**

Medlem: Nej Skriv valg år Skriv prioritet Tilføj

**Repræsentant**

Medlem: Nej 2013 Skriv sup\_t\_b Tilføj

Figur 79 - Blackbox test - Opdater medlemskab - Tilføj repræsentant

		rid	pid	sid	valg_aar	sup_t_B
<input type="checkbox"/>	Ret	154	65	41	2013	NULL

Figur 80 - Blackbox test - Opdater medlemskab - Tilføj repræsentant DB

Opdater Medlemskab

Test

**Suppleant**  
 Medlem: Nej

**Repræsentant**  
 Medlem: Ja

**Bestyrelse**  
 Medlem: Nej

Figur 81 - Blackbox test - Opdater medlemskab - Tilføj til bestyrelsen

	bid	pid	titel	valg
<input type="checkbox"/> <input type="button" value="Ret"/> <input type="button" value="Kopi"/> <input type="button" value="Slet"/>	33	65	Bestyrelsesmedlem	2013-01-01

Figur 82 - Blackbox test - Opdater medlemskab - Tilføj til bestyrelsen DB

Opdater Medlemskab

Test

**Suppleant**  
 Medlem: Nej

**Repræsentant**  
 Medlem: Ja

**Bestyrelsen**  
 Medlem: Ja

Figur 83 - Blackbox test - Fjern fra bestyrelsen

Opdater Medlemskab

Test

**Suppleant**

Medlem: Nej

**Repræsentant**

Medlem: Ja

Figur 84 - Blackbox test - Fjern fra repræsentantskab

Opdater Medlemskab

Test

**Suppleant**

Medlem: Ja

Figur 85 - Blackbox test - Fjern fra suppleant

- medlemmer						
←T→	pid	sid	valg_aar	sup_t_B	rid	
<input type="checkbox"/>	68	NULL	2000	NULL	159	

Figur 86 - Blackbox test - Opdater medlemskab - Før find knappen trykkes i repræsentant tabel

←T→	su_id	pid	sid	prioritet	valg_aar	
<input type="checkbox"/>	149	70	NULL	2	2000	
<input type="checkbox"/>	148	69	NULL	1	2000	

Figur 87 - Blackbox test - Opdater medlemskab - Før find knappen trykkes i suppleant tabel

←T→	rid	pid	sid	valg_aar	sup_t_B	
<input type="checkbox"/>	162	69	NULL	2000	1	

Figur 88 - Blackbox test - Opdater medlemskab - Efter find knappen trykkes i repræsentant tabel

←T→	su_id	pid	sid	prioritet	valg_aar	
<input type="checkbox"/>	149	70	NULL	1	2000	
<input type="checkbox"/>	154	31	NULL	2	2000	

Figur 89 - Blackbox test - Opdater medlemskab - Efter find knappen trykkes i suppleant tabel

↑ medlemslister						
←T→	bid	pid	titel	valg		
<input type="checkbox"/>	37	68	Bestyrelsesmedlem	2013-01-01		

Figur 90 - Blackbox test - Opdater medlemskab - Før find knappen trykkes i bestyrelsen

←T→	bid	pid	titel	valg	
<input type="checkbox"/>	38	69	Bestyrelsesmedlem	2013-05-24	

Figur 91 - Blackbox test - Opdater medlemskab - Efter find knappen trykkes i bestyrelsen

←T→	pid	navn	adr	pnr	tlf	mobil	e_mail	r_siden
<input type="checkbox"/>	65	Andet navn	andetvej 1	2300	87654321	87654321	andet@andet.dk	

Figur 92 - Blackbox test - Opdater person

←T→	sid	pid	adr	postnr	tlf
<input type="checkbox"/>	41	65	sommervej 1	2850	87654321

Figur 93 - Blackbox test - Opdater sommerhus

←T→	su_id	pid	sid	prioritet	valg_aar
<input type="checkbox"/>	145	65	41	2	2016

Figur 94 - Blackbox test - Opdater suppleant

←T→	rid	pid	sid	valg_aar	sup_t_B
<input type="checkbox"/>	155	65	41	2016	NULL

Figur 95 - Blackbox test - Opdater repræsentant

←T→	bid	pid	titel	valg
<input type="checkbox"/>	34	65	Formand	2016-12-12

Figur 96 - Blackbox test - Opdater bestyrelsen

←T→	sid	pid	adr	postnr	tlf
-----	-----	-----	-----	--------	-----

Figur 97 - Blackbox test - Sommerhus tabel tom

←T→	sid	pid	adr	postnr	tlf
<input type="checkbox"/>	43	65	Sommervej 1	2850	12345678

Figur 98 - Blackbox test - Opdater sommerhus data

←T→	▼	mid	pid	name	type	size	img
-----	---	-----	-----	------	------	------	-----

Figur 99 - Blackbox test - Fjern billede

←T→	▼	mid	pid	name	type	size	img					
		Ret		Kopi		Slet	185	65	Test.jpg	image/jpeg	6772	[BLOB - 6,6 KiB]

Figur 100 - Blackbox test - Tilføj billede

Tilføj Person

Suppleant:

Repræsentant:

Navn:

Suppleant og repræsentant kan ikke checkes samtidig

Figur 101 - Blackbox test - Opret rep + sup

Tilføj Person

Suppleant:

Repræsentant:

Navn:

Navn: Skal være bogstaver

Figur 102 - Blackbox test - Navn skal være bogstaver

Tilføj Person

Suppleant:

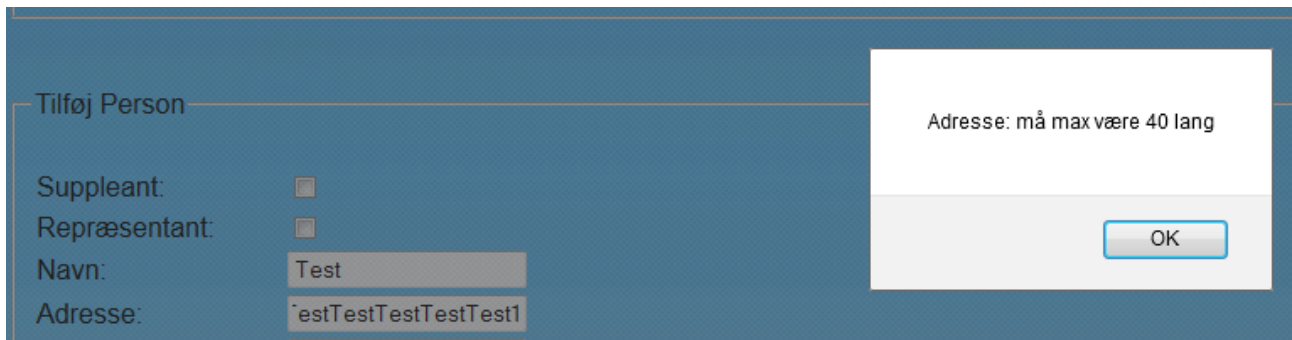
Repræsentant:

Navn:

Adresse:

Navn: må max være 30 lang

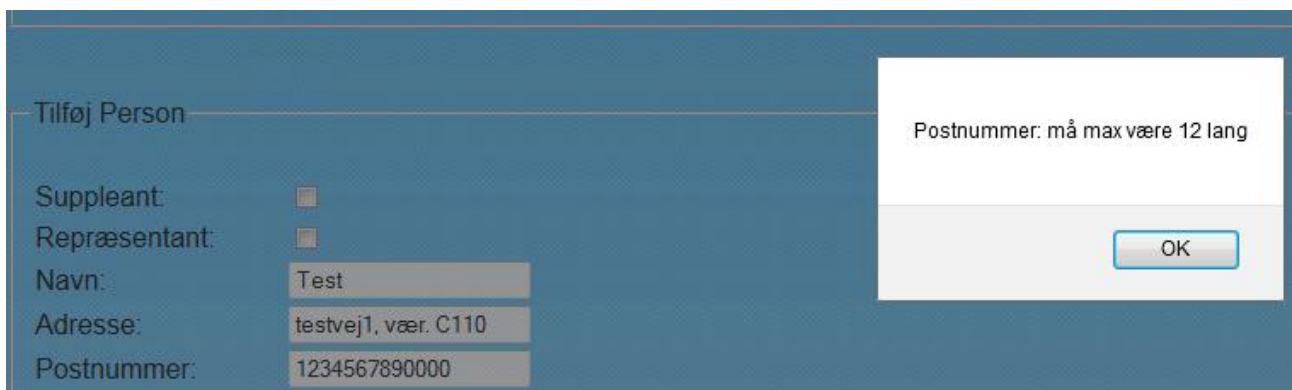
Figur 103 - Blackbox test - Navn max længde



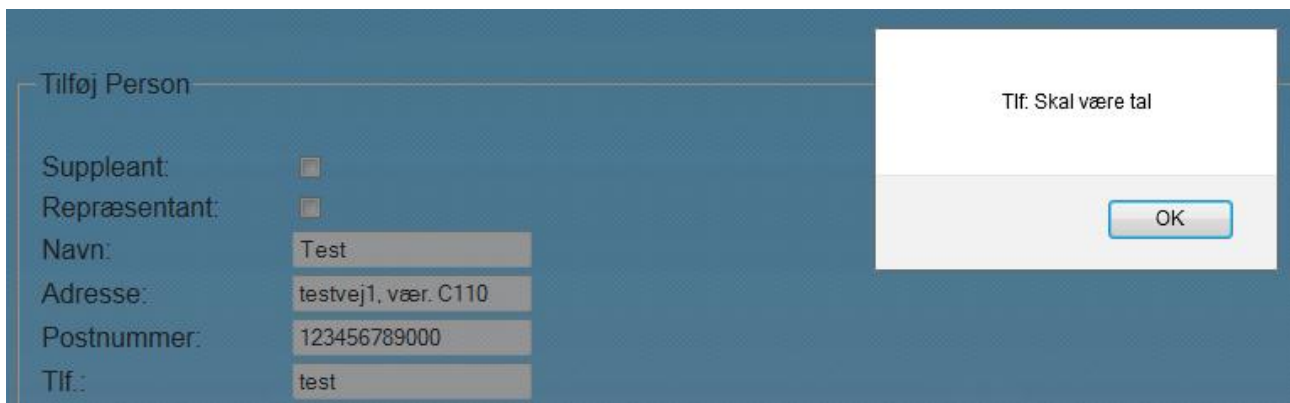
Figur 104 - Blackbox test - Adresse max længde



Figur 105 - Blackbox test - Postnummer skal være tal



Figur 106 - Blackbox test - Postnummer max længde



Figur 107 - Blackbox test - Tlf skal være tal



Figur 108 - Blackbox test - Tlf max længde

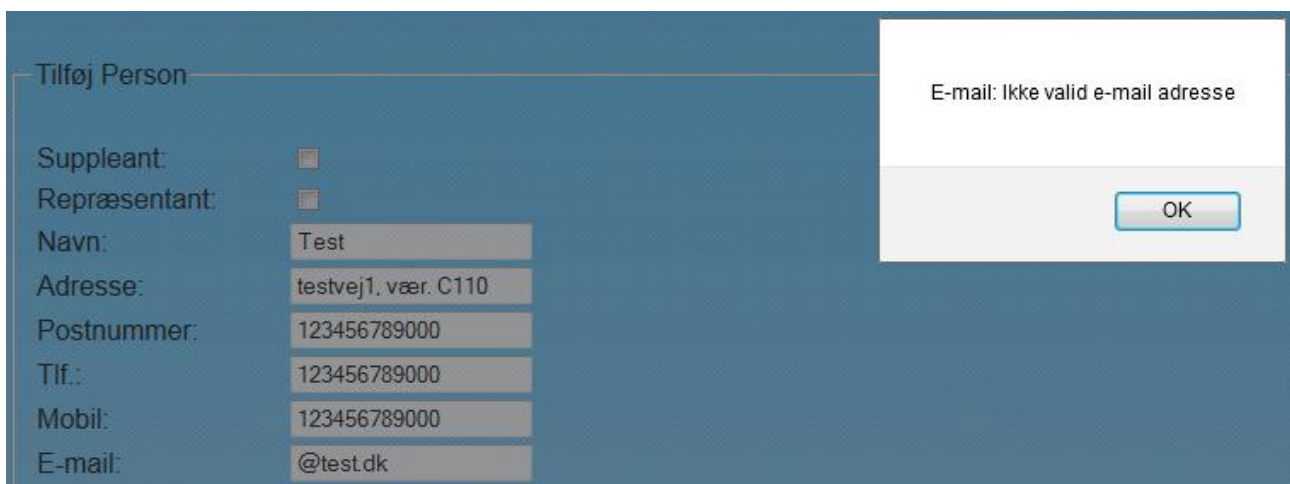


Figur 109 - Blackbox test - Mobil skal være tal





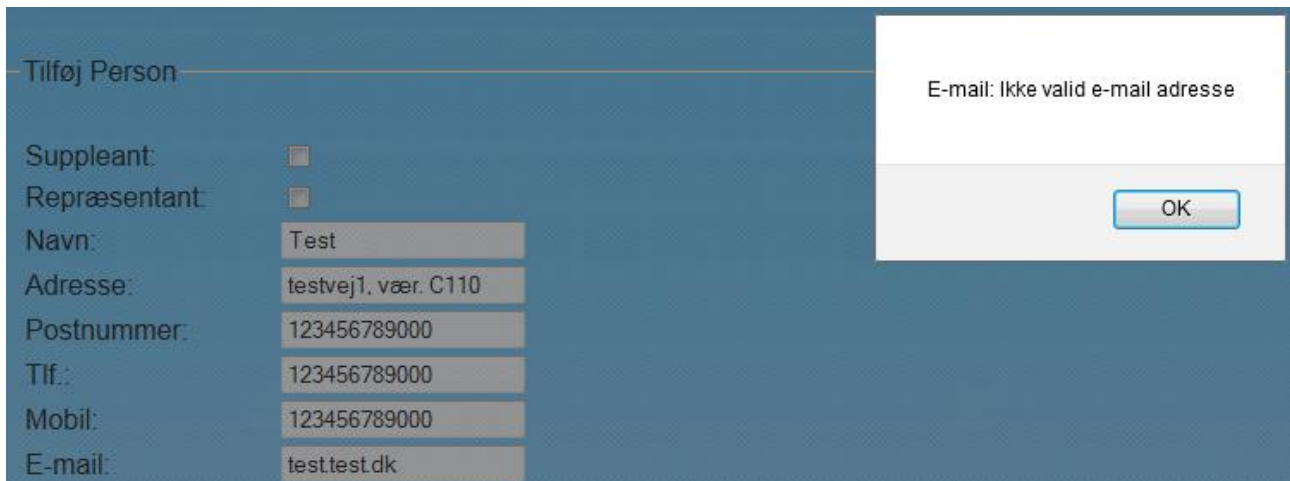
Figur 110 - Blackbox test - Mobil max længde



Figur 111 - Blackbox test - Email ikke valid



Figur 112 - Blackbox test - Email ikke valid



Figur 113 - Blackbox test - Email ikke valid



Figur 114 - Blackbox test - Suppleant valg skal være tal

Tilføj Person

Suppleant:

Repræsentant:

Navn: Test

Adresse: testvej1, vær. C110

Postnummer: 123456789000

Tlf.: 123456789000

Mobil: 123456789000

E-mail: test@test.dk

R-siden:

Sommerhus Adr.:

Sommerhus Postnr.:

Sommerhus Tlf.:

Sup Valg år: 12345

Sup Valg: må max være 4 lang

OK

Figur 115 - Blackbox test – Suppleant valg max længde

Tilføj Person

Suppleant:

Repræsentant:

Navn: Test

Adresse: testvej1, vær. C110

Postnummer: 123456789000

Tlf.: 123456789000

Mobil: 123456789000

E-mail: test@test.dk

R-siden:

Sommerhus Adr.:

Sommerhus Postnr.:

Sommerhus Tlf.:

Sup Valg år: 1234

Sup Prio: test

Sup Prio: Skal være tal

OK

Figur 116 - Blackbox test - Suppleant prioritet skal være tal

Tilføj Person

Suppleant:	<input checked="" type="checkbox"/>
Repræsentant:	<input type="checkbox"/>
Navn:	Test
Adresse:	testvej1, vær. C110
Postnummer:	123456789000
Tlf.:	123456789000
Mobil:	123456789000
E-mail:	test@test.dk
R-siden:	
Sommerhus Adr.:	
Sommerhus Postnr.:	
Sommerhus Tlf.:	
Sup Valg år:	1234
Sup Prio:	123

Sup Prio: må max være 2 lang

OK

Figur 117 - Blackbox test - Suppleant prioritet max længde

Tilføj Person

Suppleant:	<input type="checkbox"/>
Repræsentant:	<input checked="" type="checkbox"/>
Navn:	Test
Adresse:	testvej1, vær. C110
Postnummer:	123456789000
Tlf.:	123456789000
Mobil:	123456789000
E-mail:	test@test.dk
R-siden:	
Sommerhus Adr.:	
Sommerhus Postnr.:	
Sommerhus Tlf.:	
Rep Valg år:	test

Rep Valg: Skal være tal

OK

Figur 118 - Blackbox test - Repræsentant valg skal være tal

Tilføj Person

Suppleant:

Repræsentant:

Navn:

Adresse:

Postnummer:

Tlf.:

Mobil:

E-mail:

R-siden:

Sommerhus Adr.:

Sommerhus Postnr.:

Sommerhus Tlf.:

Rep Valg år:

Rep Valg: må max være 4 lang

OK

Figur 119 - Blackbox test - Repræsentant valg max længde

R-siden:

Sommerhus Adr.:

Sommerhus Postnr.:

Sommerhus Tlf.:

Tilføj

Gennemse\_ upload

Vælg en person som skal slettes

OK

Slet Person Fra Systemet

Vælg en person:  Slet

Figur 120 - Blackbox test - Slet en person

Slet Person Fra Systemet

Vælg en person:  Slet

Opdater Medlemskab

Test

**Suppleant**  
 Medlem: Ja

**Repræsentant**  
 Medlem: Nej

**Bestyrelse**  
 Medlem: Nej

Fjern fra Sup

OK

Figur 121 - Blackbox test - Tilføj repræsentant

Vælg en person:  Slet

Opdater Medlemskab

Test

**Suppleant**  
 Medlem: Ja

**Repræsentant**  
 Medlem: Nej

**Bestyrelse**  
 Medlem: Nej

Fjern fra Sup + Skal være med i Rep

OK

Figur 122 - Blackbox test - Tilføj til bestyrelsen



Opdater Medlemskab

Test

**Suppleant**  
 Medlem: Nej 2013 1 Tilføj

**Repræsentant**  
 Medlem: Ja Fjern REP

**Bestyrelse**  
 Medlem: Nej YEAR-MM-DD Vælg titel Tilføj

**Erstat person**  
 Erstatte rep/bes: Find

Fjern fra Rep/Bes

OK

Figur 123 - Blackbox test - Tilføj til suppleant

Slet Person Fra Systemet

Vælg en person: Slet

Fjern fra Sup + Skal være med i Rep

OK

Opdater Medlemskab

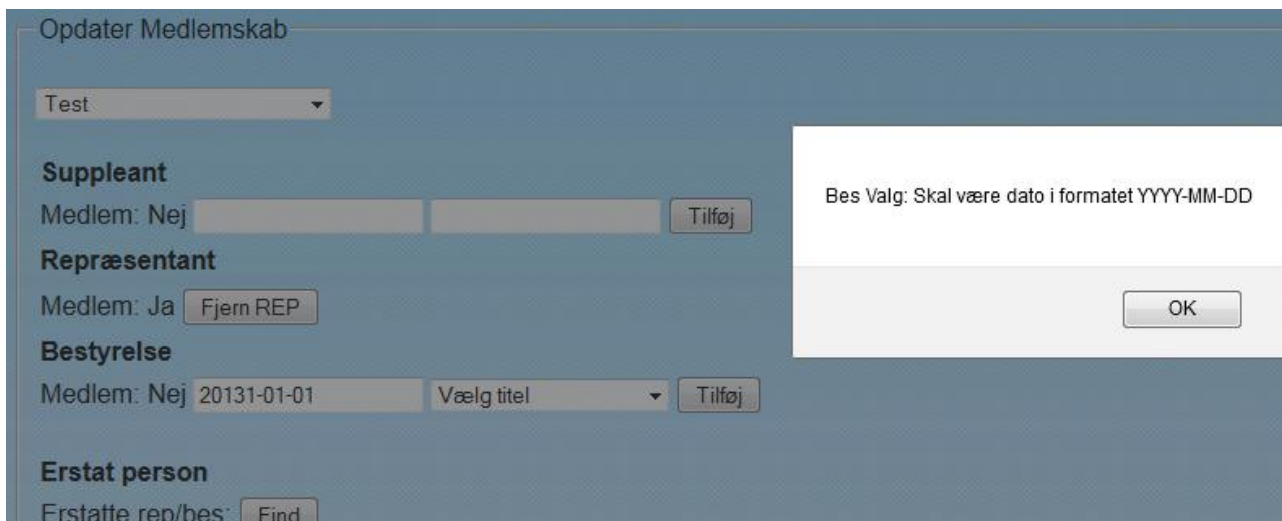
Test

**Suppleant**  
 Medlem: Nej Skriv valg år Skriv prioritet Tilføj

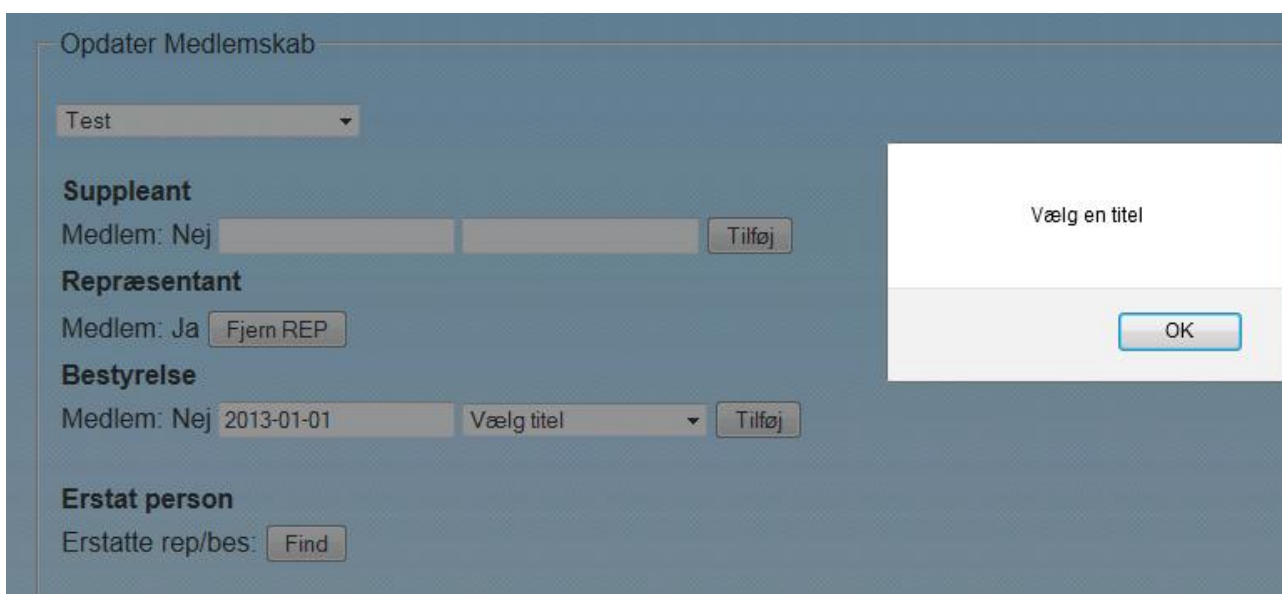
**Repræsentant**  
 Medlem: Nej Skriv valg år Skriv sup\_t\_b Tilføj

**Bestyrelse**  
 Medlem: Nej 2013-01-01 Bestyrelsesmedlem Tilføj

Figur 124 - Blackbox test -Tilføj til bestyrelsen



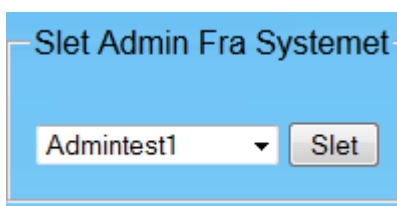
Figur 125 - Blackbox test - Tilføj til bestyrelsen - Skal være i bestemt format



Figur 126 - Blackbox test - Tilføj til bestyrelsen - Vælg en titel

	aid	username	password	email
Ret	4	Admintest1	Topsecret1	AdminTest@test.dk

Figur 127 - Blackbox test - Opret admin



Figur 128 - Blackbox test - Slet admin



	aid	username	password	email
Ret	4	Admintest2	Topsecret2	AdminTest2@test.dk

Figur 129 - Blackbox test - Rediger Admin

Opret Admin

Brugernavn:

E-mail:

Kodeord:

Admin Brugernavn: Skal være bogstaver og tal

Figur 130 - Blackbox test - Brugernavn skal være bogstaver og tal

Opret Admin

Brugernavn:

E-mail:

Kodeord:

Admin Brugernavn: må max være 30 lang

Figur 131 - Blackbox test - Brugernavn max længde

Opret Admin

Brugernavn:

E-mail:

Kodeord:

Admin Email: Ikke valid e-mail adresse

Figur 132 - Blackbox test - Email ikke valid

Opret Admin

Brugernavn: Testbruger1Testbruger

E-mail: test@dk

Kodeord:

Tilføj

Admin Email: Ikke valid e-mail adresse

OK

Figur 133 - Blackbox test - Email ikke valid

Opret Admin

Brugernavn: Testbruger1Testbruger

E-mail: admin.test.dk

Kodeord:

Tilføj

Admin Email: Ikke valid e-mail adresse

OK

Figur 134 - Blackbox test - Email ikke valid

Opret Admin

Brugernavn: Testbruger1Testbruger

E-mail: admin@test.dk

Kodeord: Forkort

Tilføj

Kodeord skal være mindst 8 karakterer lang

OK

Figur 135 - Blackbox test - Kodeord minimum længde

**Login**

Brugernavn :

Kodeord :

[Glemt Kodeord](#)

Login

Brugernavn/Kodeord er forkert

Figur 136 - Blackbox test – Login

Brugernavn:

Figur 137 - Blackbox test - Glemte kodeord

Brugernavn:    
 Brugernavn blev ikke fundet

Figur 138 - Blackbox test - Brugernavn ikke fundet

**W3C** Markup Validation Service  
 Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed		
<b>Address :</b>	<input type="text" value="http://www.rorvigvand.dk/index.php"/>		
<b>Modified:</b>	(undefined)		
<b>Server:</b>	Apache		
<b>Size:</b>	4651		
<b>Content-Type:</b>	text/html		
<b>Encoding :</b>	utf-8	<input type="button" value="(detect automatically)"/>	
<b>Doctype :</b>	HTML 4.01 Transitional	<input type="button" value="(detect automatically)"/>	
<b>Root Element:</b>	html		

Figur 139 - HTML valideringstest - Index

**W3C** Markup Validation Service  
 Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed		
<b>Address :</b>	<input type="text" value="http://www.rorvigvand.dk/kontaktpersoner.php"/>		
<b>Modified:</b>	(undefined)		
<b>Server:</b>	Apache		
<b>Size:</b>	6904		
<b>Content-Type:</b>	text/html		
<b>Encoding :</b>	utf-8	<input type="button" value="(detect automatically)"/>	
<b>Doctype :</b>	HTML 4.01 Transitional	<input type="button" value="(detect automatically)"/>	
<b>Root Element:</b>	html		

Figur 140 - HTML valideringstest - Kontaktpersoner



## Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Congratulations](#) · [Icons](#) [Source Listing](#) [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed		
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/webmaster.php"/>		
<b>Modified:</b>	(undefined)		
<b>Server:</b>	Apache		
<b>Size:</b>	4929		
<b>Content-Type:</b>	text/html		
<b>Encoding:</b>	utf-8	<input type="text" value="(detect automatically)"/>	▼
<b>Doctype:</b>	HTML 4.01 Transitional	<input type="text" value="(detect automatically)"/>	▼
<b>Root Element:</b>	html		

Figur 141 - HTML valideringstest - Webmaster



## Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Congratulations](#) · [Icons](#) [Source Listing](#) [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed		
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/bestyrelsen.php"/>		
<b>Modified:</b>	(undefined)		
<b>Server:</b>	Apache		
<b>Size:</b>	70131		
<b>Content-Type:</b>	text/html		
<b>Encoding:</b>	utf-8	<input type="text" value="(detect automatically)"/>	▼
<b>Doctype:</b>	HTML 4.01 Transitional	<input type="text" value="(detect automatically)"/>	▼
<b>Root Element:</b>	html		

Figur 142 - HTML valideringstest - Bestyrelsen



## Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Congratulations](#) · [Icons](#) [Source Listing](#) [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed		
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/medlemmer.php"/>		
<b>Modified:</b>	(undefined)		
<b>Server:</b>	Apache		
<b>Size:</b>	10814		
<b>Content-Type:</b>	text/html		
<b>Encoding:</b>	utf-8	<input type="text" value="(detect automatically)"/>	▼
<b>Doctype:</b>	HTML 4.01 Transitional	<input type="text" value="(detect automatically)"/>	▼
<b>Root Element:</b>	html		

Figur 143 - HTML valideringstest - Medlemmer

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/suppleanter.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	7330
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 144 - HTML valideringstest - Suppleanter

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/repraesentantskab.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4738
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 145 - HTML valideringstest - Repræsentantskab

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/historie.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	6567
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 146 - HTML valideringstest - Historie

[Jump To:](#)
[Congratulations - Icons](#)
[Source Listing](#)
[Outline](#)

This document was successfully checked as HTML 4.01 Transitional!	
<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/login.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	5765
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 147 - HTML valideringstest - Login

[Jump To:](#)
[Congratulations - Icons](#)
[Source Listing](#)
[Outline](#)

This document was successfully checked as HTML 4.01 Transitional!	
<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/email.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4795
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 148 - HTML valideringstest - Email

[Jump To:](#)
[Congratulations - Icons](#)

This document was successfully checked as HTML 4.01 Transitional!	
<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/meddelelser.php"/>
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 149 - HTML valideringstest - Meddelelser

Jump To: [Congratulations](#) · [Icons](#)

This document was successfully checked as HTML 4.01 Transitional!	
<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://www.rorvigvand.dk/moedeplan.php"/>
<b>Encoding:</b>	utf-8 <input type="button" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="button" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 150 - HTML valideringstest - Mødeplan

Jump To: [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!	
<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://rorvigvand.dk/sadmin.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4428
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="button" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="button" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 151 - - HTML valideringstest - Sadmin

Jump To: [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!	
<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://rorvigvand.dk/sbestyrelsen.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4420
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="button" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="button" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 152 - HTML valideringstest - Sbestyrelsen

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

[Jump To:](#) [Congratulations - Icons](#) [Source Listing](#) [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://zorvigvand.dk/sbilleder.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4429
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 153 - HTML valideringstest - Sbilleder

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

[Jump To:](#) [Congratulations - Icons](#) [Source Listing](#) [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://zorvigvand.dk/sperson.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4428
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 154 - HTML valideringstest - Sperson

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

[Jump To:](#) [Congratulations - Icons](#) [Source Listing](#) [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://zorvigvand.dk/srepraesentant.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4433
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 155 - HTML valideringstest - Srepraesentant



**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Congratulations](#) - [Icons](#) - [Source Listing](#) - [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://rorvigvand.dk/ssommerhus.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4429
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 156 - HTML valideringstest - Sommerhus

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Congratulations](#) - [Icons](#) - [Source Listing](#) - [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://rorvigvand.dk/ssuppleant.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	4429
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 157 - HTML valideringstest - Ssupleant

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Congratulations](#) - [Icons](#) - [Source Listing](#) - [Outline](#)

This document was successfully checked as HTML 4.01 Transitional!

<b>Result:</b>	Passed
<b>Address:</b>	<input type="text" value="http://rorvigvand.dk/admin_management.php"/>
<b>Modified:</b>	(undefined)
<b>Server:</b>	Apache
<b>Size:</b>	5251
<b>Content-Type:</b>	text/html
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	HTML 4.01 Transitional <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	html

Figur 158 - HTML valideringstest - Admin management

**W3C** Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Congratulations](#) · [Icons](#) · [Source Listing](#) · [Outline](#)

**This document was successfully checked as HTML 4.01 Transitional!**

<b>Result:</b>	Passed		
<b>Address :</b>	<input type="text" value="http://rorvigvand.dk/user_management.php"/>		
<b>Modified:</b>	(undefined)		
<b>Server:</b>	Apache		
<b>Size:</b>	8190		
<b>Content-Type:</b>	text/html		
<b>Encoding :</b>	utf-8	<input type="text" value="(detect automatically)"/>	
<b>Doctype :</b>	HTML 4.01 Transitional	<input type="text" value="(detect automatically)"/>	
<b>Root Element:</b>	html		

Figur 159 - HTML valideringstest - User management

**W3C** The W3C CSS Validation Service  
W3C CSS Validator results for <http://www.rorvigvand.dk/index.php> (CSS level 3)

**Jump to:** [Warnings \(6\)](#) · [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/index.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 160 - CSS valideringstest - Index

**W3C** The W3C CSS Validation Service  
W3C CSS Validator results for <http://www.rorvigvand.dk/kontaktpersoner.php> (CSS level 3)

**Jump to:** [Warnings \(6\)](#) · [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/kontaktpersoner.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 161 - CSS valideringstest - Kontaktpersoner

**W3C** The W3C CSS Validation Service  
W3C CSS Validator results for <http://www.rorvigvand.dk/webmaster.php> (CSS level 3)

**Jump to:** [Warnings \(6\)](#) · [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/webmaster.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 162 - CSS valideringstest - Webmaster



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/bestyrelsen.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/bestyrelsen.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 163 - CSS valideringstest - Bestyrelsen



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/medlemmer.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/medlemmer.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 164 - CSS valideringstest - Medlemmer



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/suppleanter.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/suppleanter.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 165 - CSS valideringstest - Suppleanter



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/repraesentantskab.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/repraesentantskab.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 166 - CSS valideringstest - Repræsentantskab



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/historie.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/historie.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 167 - CSS valideringstest - Historie



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/login.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/login.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#) !

### Figur 168 - CSS valideringstest - Login



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/email.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/email.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#) !

### Figur 169 - CSS valideringstest - Email



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/meddelelser.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/meddelelser.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#) !

### Figur 170 - CSS valideringstest - Meddelelser



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/moedeplan.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/moedeplan.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#) !

### Figur 171 - CSS valideringstest - Mødeplan



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/sadmin.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/sadmin.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#) !

### Figur 172 - CSS valideringstest - Sadmin



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/sbestyrelsen.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/sbestyrelsen.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 173 - CSS valideringstest - Sbestyrelsen



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/sbilleder.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/sbilleder.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 174 - CSS valideringstest - Sbilleder



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/sperson.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/sperson.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 175 - CSS valideringstest - Sperson



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/srepraesentant.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/srepraesentant.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 176 - CSS valideringstest - Srepraesentant



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/ssommerhus.php> (CSS level 3)

Jump to: [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/ssommerhus.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 177 - CSS valideringstest - Ssommerhus



## The W3C CSS Validation Service

W3C CSS Validator results for <http://www.rorvigvand.dk/ssuppleant.php> (CSS level 3)

[Jump to:](#) [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for <http://www.rorvigvand.dk/ssuppleant.php> (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 178 - CSS valideringstest - Ssuppleant



## The W3C CSS Validation Service

W3C CSS Validator results for [http://www.rorvigvand.dk/admin\\_management.php](http://www.rorvigvand.dk/admin_management.php) (CSS level 3)

[Jump to:](#) [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for [http://www.rorvigvand.dk/admin\\_management.php](http://www.rorvigvand.dk/admin_management.php) (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 179 - CSS valideringstest - Admin management



## The W3C CSS Validation Service

W3C CSS Validator results for [http://www.rorvigvand.dk/user\\_management.php](http://www.rorvigvand.dk/user_management.php) (CSS level 3)

[Jump to:](#) [Warnings \(6\)](#) [Validated CSS](#)

W3C CSS Validator results for [http://www.rorvigvand.dk/user\\_management.php](http://www.rorvigvand.dk/user_management.php) (CSS level 3)

**Congratulations! No Error Found.**

This document validates as [CSS level 3](#)!

Figur 180 - CSS valideringstest - User management

## 19.7. Browsertest



Figur 181 - Browsertest - Win XP – IE 8.0



Figur 182 - Browsertest - Win XP - Firefox 20





Figur 183 - Browsertest - Win XP - Chrome 26



Figur 184 - Browsertest - Win XP - Safari 5.1



Figur 185 - Browsertest - Win 7 - Explorer 10



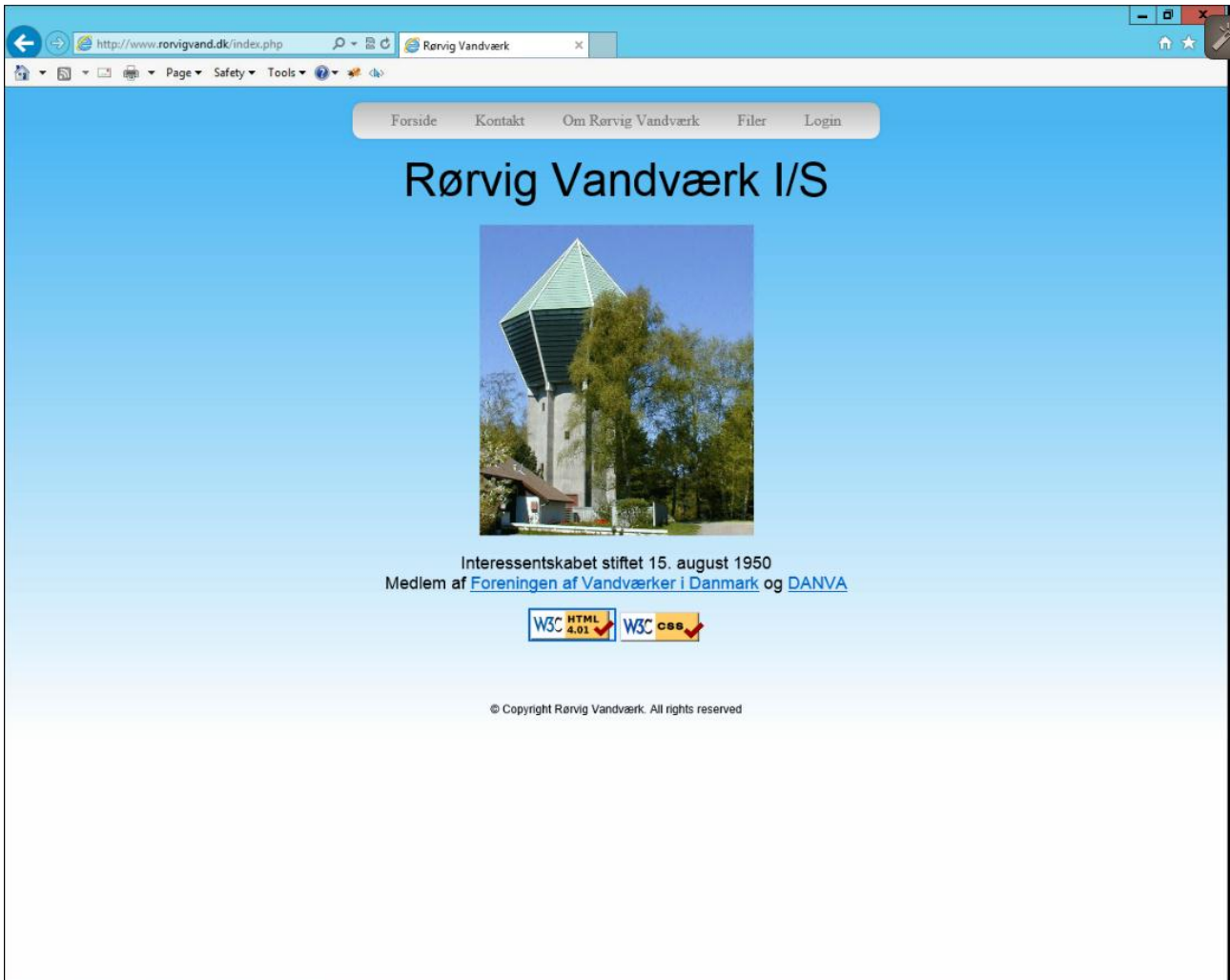
Figur 186 - Browsertest - Win 7 - Firefox 20



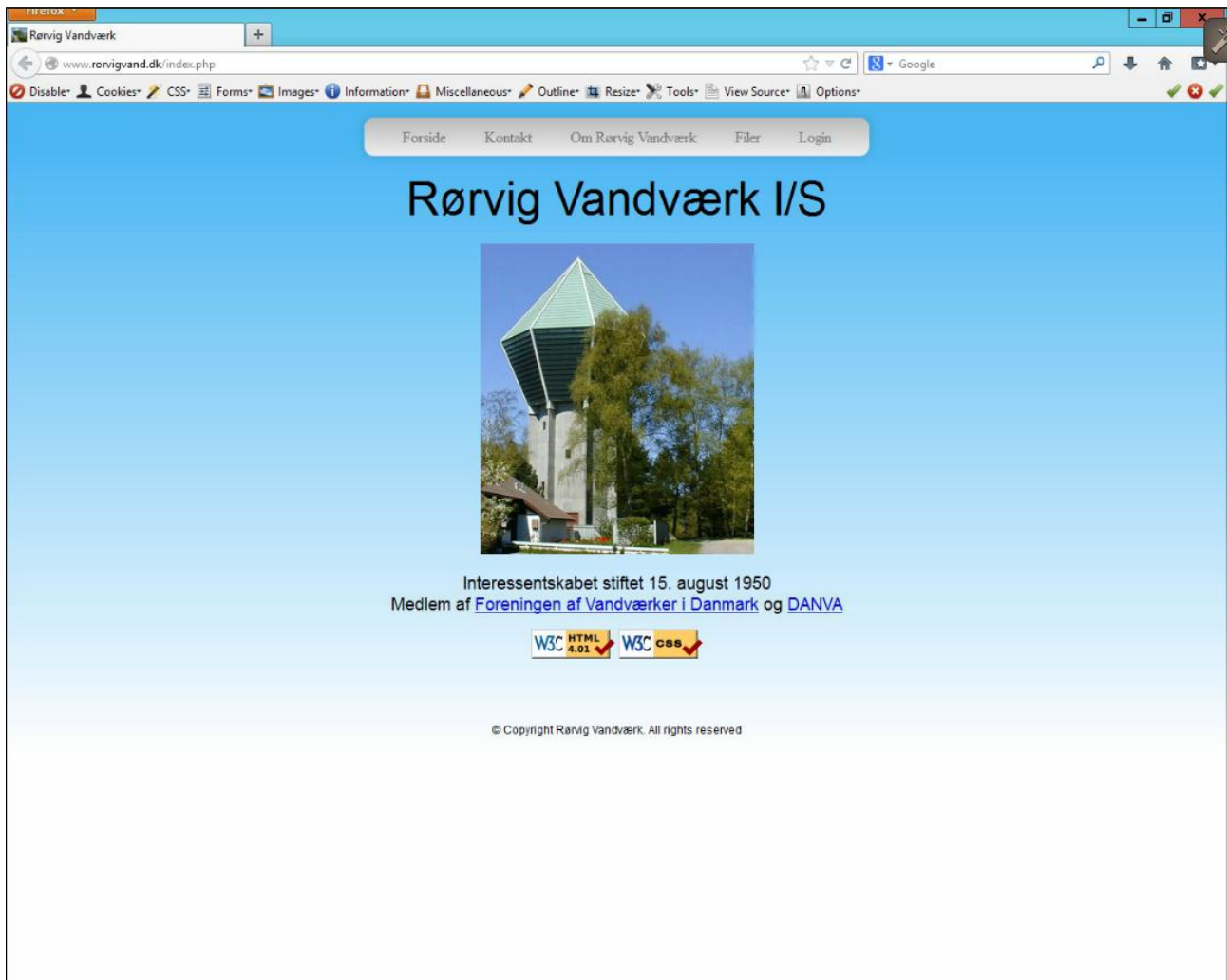
Figur 187 - Browsertest - Win 7 - Chrome 26



Figur 188 - Browsertest - Win 7 - Safari 5.1



Figur 189 - Browsertest - Win 8 - Explorer 10



Figur 190 - Browsertest - Win 8 - Firefox 20

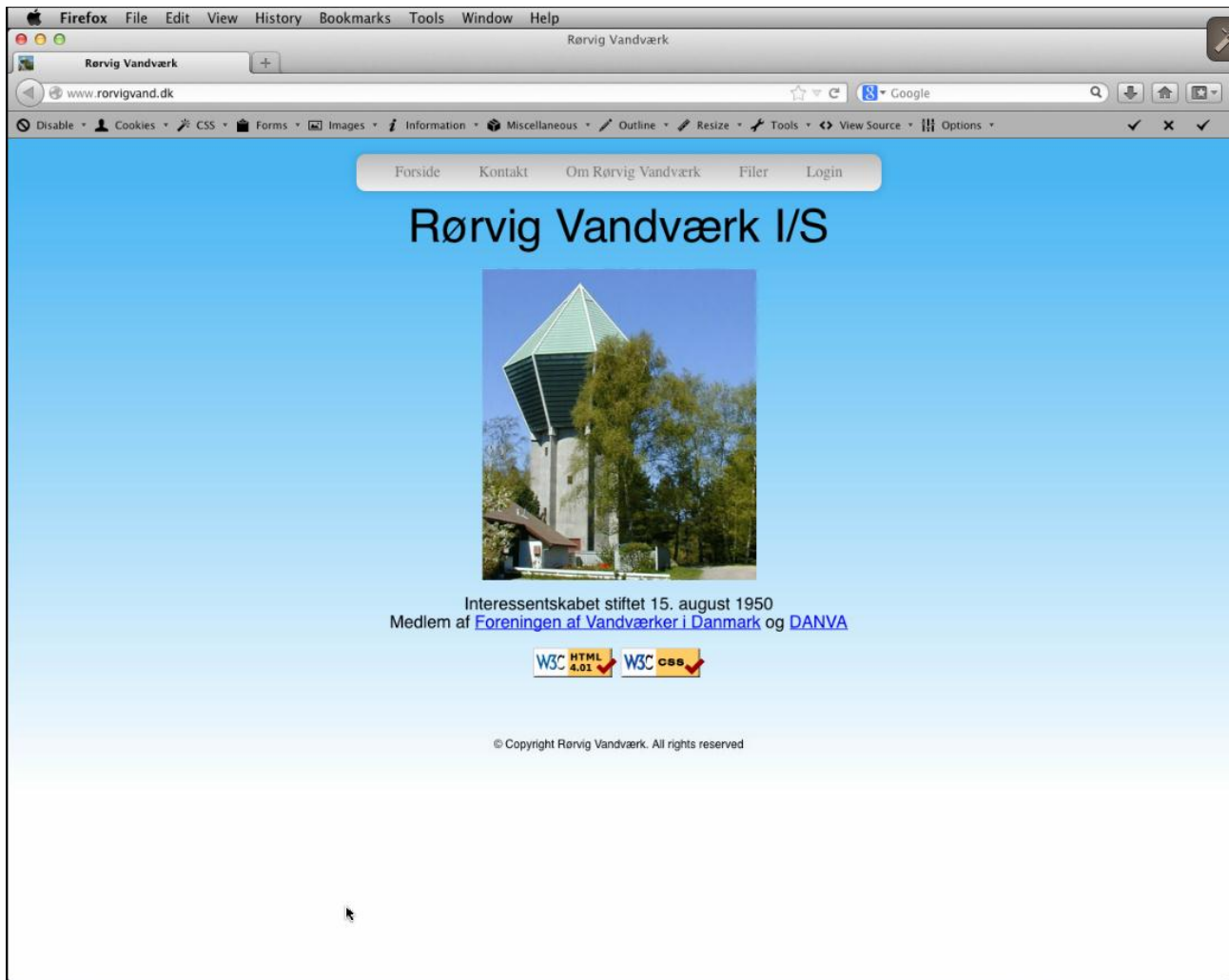




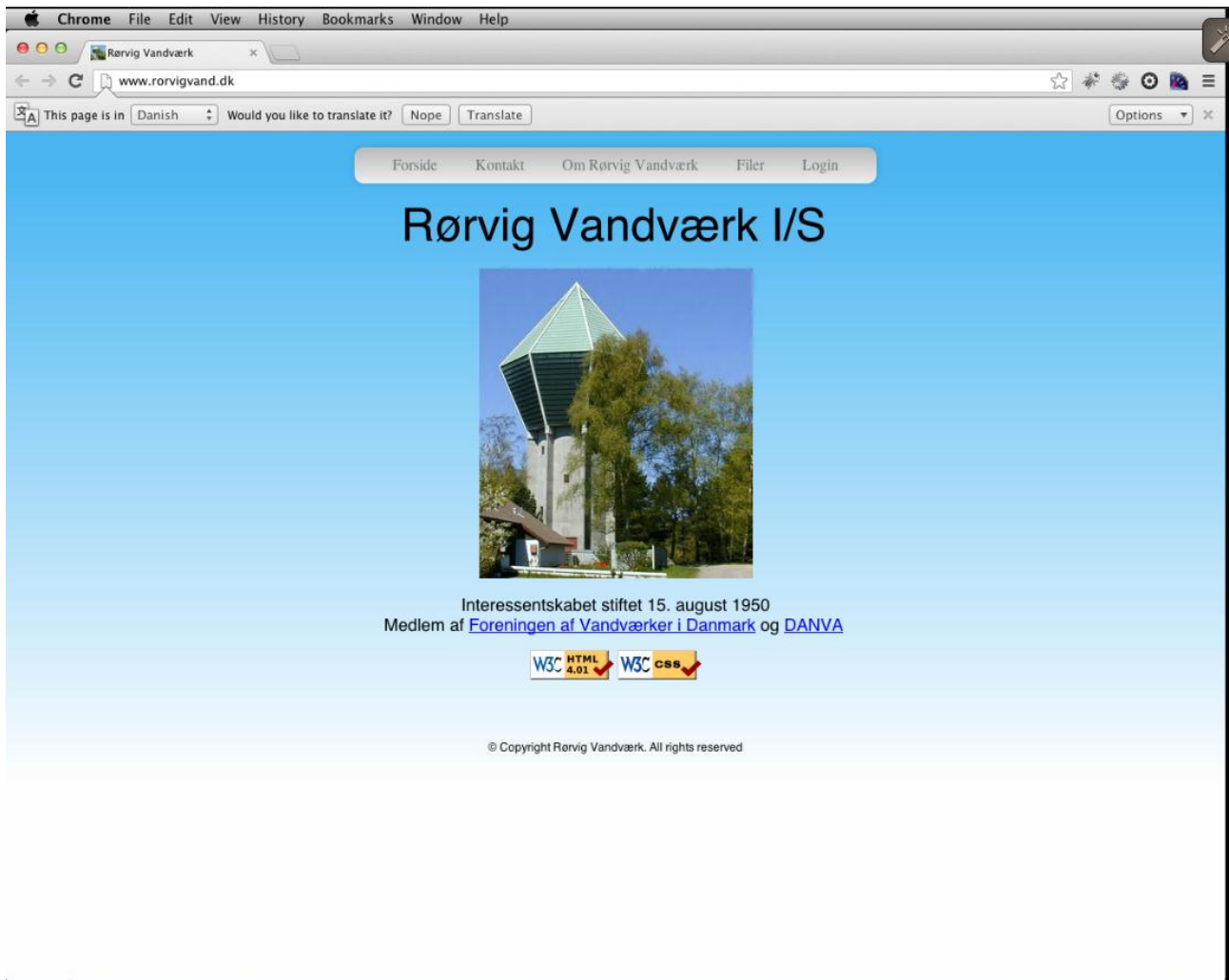
Figur 191 - Browsertest - Win 8 - Chrome 26



Figur 192 - Browsertest - Win 8 - Safari 5.1



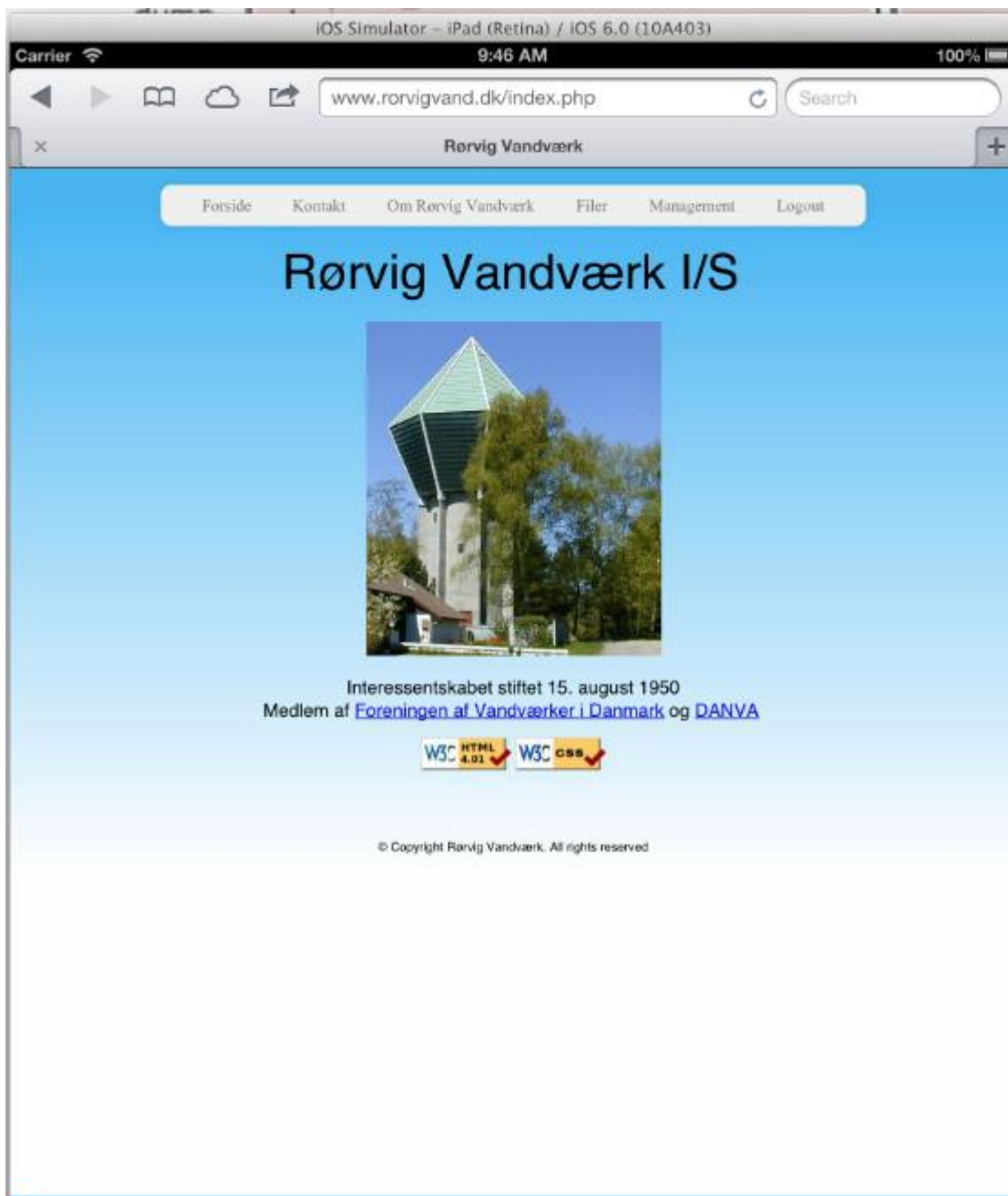
Figur 193 - Browsertest - OSX Lion - Firefox 20



Figur 194 - Browsertest - OS X Lion - Chrome 26



Figur 195 - Browsertest - OSX Lion - Safari 6



Figur 196 - Browsertest - IOS 6 - iPad3 Safari

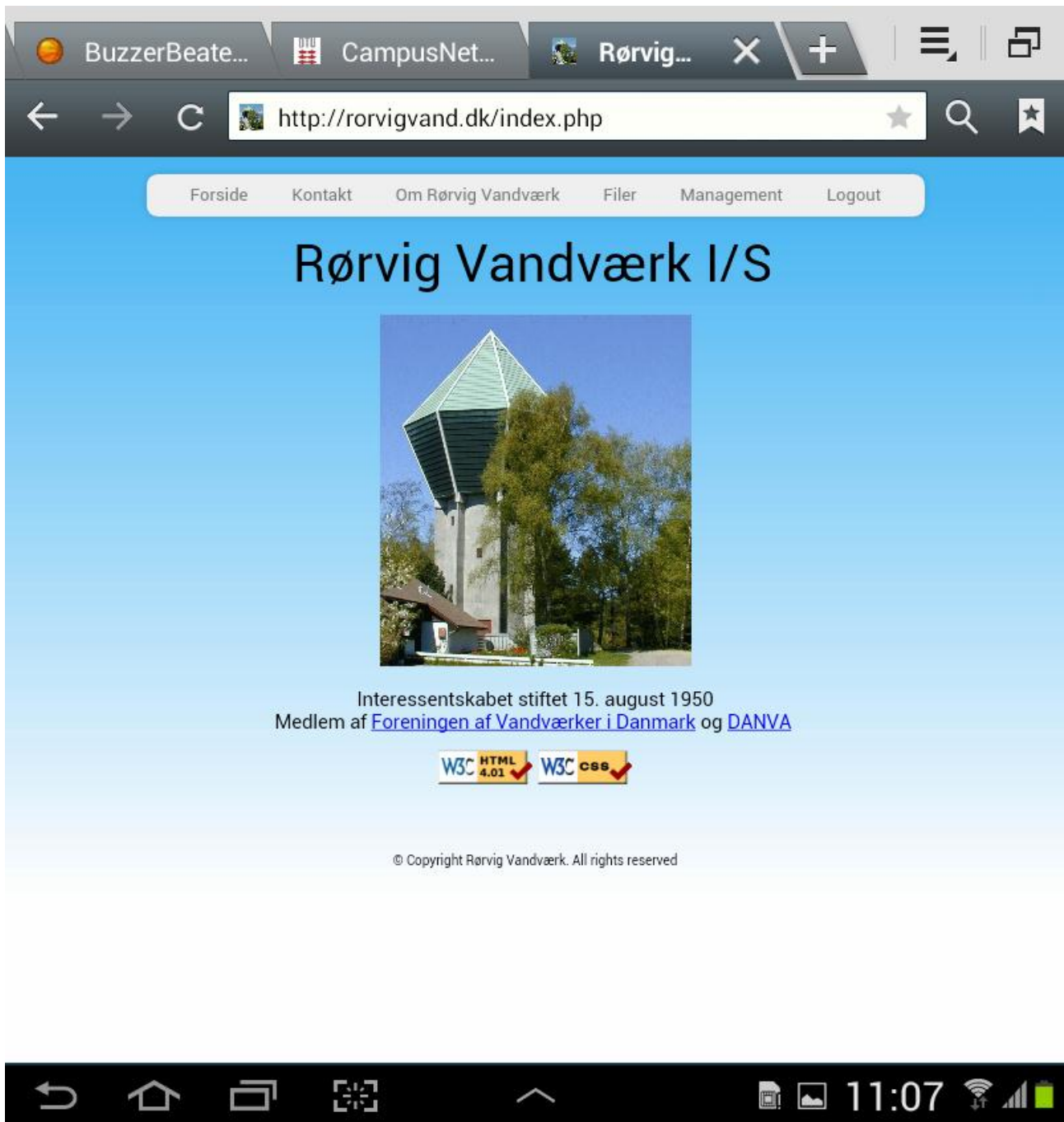


Figur 197 - Browsertest - IOS 6 - iPhone 5 Safari



Figur 198 - Browsertest - Android 4.1- Samsung Galaxy SIII





Figur 199 - Browsertest - Android 4.1 - Samsung Galaxy Note 10.1



## Rørvig Vandværk I/S

Undgå ubehagelige overraskelser. Husk at lukke for stophanen når sommerhuset forlades efter hver weekend

[Hovedside](#)

[Meddelelser](#)

[Advarsel](#)

[Analyser](#)

[Historie](#)

[Vedtægter](#)

[Taksblad](#)

[Regulativ](#)

[Bestyrelse](#)

[Kontakt](#)

[Repræsentantskab](#)

[Webmaster](#)

# Velkommen til hjemmesiden for Rørvig Vandværk I/S



Interessentskabet stiftet 15. august 1950  
Medlem af Foreningen af Vandværker i Danmark og DANVA

Opdateret: Wednesday 23-01-2013 13:16:38.

Der har været 7496 besøg siden 18.02.07



[Design egen hjemmeside](#)

One.com

Nemt og hurtigt med One.com Over 1 mio. tilfredse kunder

Annoncevalg >

Figur 200 - Screenshot - Gammel webside