Udvikling af en web-baseret licensmanager

Skrevet af: Esbern Andersen-Hoppe - s093484
Vejleder: Stig Høgh
Rapport nummer: IMM-B.Eng-2012-48

June 27, 2013

Abstract

Firmaet CapNordic A/S arbejder med elektronisk dokument behandling. Kunder kan købe deres produkt, som hjælper med at holde styr på fakturaer, plus de kan købe ekstra moduler alt efter hvad de har brug for at kunne gøre med fakturaerne. CapNordic A/S laver 3 licenser til produktet, den ene er en kunde-licens så produktet kan bruges af kunden. Den anden licens er en vedligeholdelses-licens der angiver hvor lang tid produktet er gyldigt hos en kunde og hvornår de skal have fornyet produktet hvis de fortsat vil bruge det. Den sidste er en modul-licens, den kan låse op for forskellige moduler til produktet som kunderne kan vælge at købe. Disse 3 filer bliver på nuværende tidspunkt lavet på en bestemt computer på kontoret og filerne skal sendes manuelt til kunderne. For at gøre det nemmere og mindske det manuelle arbejde, så kunne CapNordic A/S godt tænke sig en web-baseret cloud-løsning. Så i stedet for at man skal have adgang til en bestemt computer for at lave licenserne, så kan de laves uanset hvor man er. Der udover så vil det være nemmere hvis licens-filerne kunne blive lagt ned på en server/cloud, som produktet vil have adgang til og så selv kunne hente filerne ind, eller så kunderne kan få fat i filerne og indlæse dem i produktet.

Contents

1 Krav og indledende tanker **1**
- 1.1 Krav fra CapNordic A/S . . . 1
- 1.2 Indledende tanker til projektet . . . 2

2 CapNordic A/S **4**
- 2.1 Om CapNordic A/S . . . 4
- 2.2 CapNordic's produkt . . . 4
  - 2.2.1 Workflow V5 . . . 5
  - 2.2.2 Workflow V8 . . . 5
- 2.3 Licensfiler . . . 5
  - 2.3.1 Hovedlicens/Kundelicens fil . . . 5
  - 2.3.2 Vedligeholdelses fil . . . 6
  - 2.3.3 Modul fil . . . 6

3 Udvikling **7**
- 3.1 Den web-baserede licensmanager . . . 7
  - 3.1.1 Indhold i web-delen . . . 7
  - 3.1.2 Visual Basic klasser . . . 8
- 3.2 Udvikling af Visual Basic klasser . . . 9
  - 3.2.1 Connection.VB . . . 9
  - 3.2.2 DbFunctions.VB . . . 10
  - 3.2.3 CreateFile.VB . . . 11
  - 3.2.4 Fejl håndtering i Visual Basic klasserne . . . 16
- 3.3 Udvikling af web-delen . . . 17
  - 3.3.1 Site.Master . . . 17
  - 3.3.2 Style1.CSS . . . 17
  - 3.3.3 Login.ASPX . . . 19
  - 3.3.4 Frontpage.ASPX . . . 21
  - 3.3.5 License.ASPX . . . 26
  - 3.3.6 Maintenance.ASPX . . . 28
  - 3.3.7 Modules.ASPX . . . 29
  - 3.3.8 Download.ASPX . . . 34
  - 3.3.9 Ajax Control toolkit . . . 35
  - 3.3.10 Fejlhåndtering i web-delen . . . 36

4 Test **39**
- 4.1 Test af Visual Basic klasserne . . . 39
  - 4.1.1 DbFunctions.VB test . . . 39
  - 4.1.2 Connection.VB test . . . 40
  - 4.1.3 CreatFile.VB . . . 40
- 4.2 Test af web-delen . . . 41
  - 4.2.1 Login.ASPX test . . . 41

4.2.2 Frontpage.ASPX test 41
4.2.3 License.ASPX test 43
4.2.4 Maintenance.ASPX test 43
4.2.5 Module.ASPX test 43
4.2.6 Download.ASPX test 45
4.2.7 Test af fejlhåndtering 45
4.3 Test i andre browsere 46
4.3.1 Mozilla Firefox 46
4.3.2 Internet Explorer 46

5 Konklusion **47**
Indholdsfortegnelse

1 Krav og indledende tanker

I dette kapitel vil jeg komme ind på hvilke krav der har været omkring den web-baserede licensmanager, hvad det endelige produkt skal kunne og hvad mine indledende tanker har været, før jeg kunne gå i gang med at udvikle den web-baserede licensmanager.

1.1 Krav fra CapNordic A/S

- Det skal være en web-løsning, så alle licenser kan laves over nettet, uanset hvor man sidder henne. Dette er nødvendigt, da det ikke altid vil være muligt at få lavet fjern-adgang til den computer der har den nuværende licensmanager. Der udover skal den være web-baseret, så alle konsulenter også vil have adgang til hjemmesiden og ikke har brug for at få kontakt til en af cheferne og få dem til at lave licenserne. Dette gør at konsulenterne hurtigt kan få lavet en licens til en kunde og få givet licensen til kunden, i stedet for at de skal vente på at en anden har tid til at få dem lavet og sendt ud til kunden.

- Den web-baserede licensmanager skal kunne lave licens-filer til CapNordic Workflow version 8 og version 5, det er nødvendigt, da der er nogle kunder der bruger den gamle version og nogle der bruger den nye version, plus det er muligt for et nyt firma at købe enten Workflow version 5 eller Workflow version 8. Så der skal kunne laves filer til begge produkterne. Grunden til at filerne ikke kan laves så de virker uanset hvilket produkt der bliver brugt, er fordi Workflow version 5 er udviklet i VB6 og Workflow version 8 er udviklet i VB.NET. Disse 2 udviklingssprog indlæser filer på forskellige måder, hvilket gør at den encoding der skal bruges, når filerne bliver genereret, bliver nødt til at være forskellig alt efter om den licens-fil der skal laves, skal bruges til version 5 eller version 8.

- De licens-filer der bliver lavet, skal kunne virke sammen med de licens-filer der kan laves fra den gamle licensmanager. Det vil sige, hvis et produkt er blevet sat op med en hovedlicens-fil fra den gamle licensmanager, så skal det være muligt at kunne indlæse fornyelses licens-filer og modul licens-filer fra den web-baserede licensmanager, uden at det fejler og at produktet så ikke virker mere. Det samme skal også være gældende i den omvendte situation, hvor hovedlicens-filen kommer fra den web-baserede licensmanager.

- Når en fil er blevet lavet, så skal den gemmes på CapNordic A/S' server, således at kundernes produkt kan få fat i licens-filerne. Filerne skal gemmes i en bestemt mappe, til det enkelt firma. På et senere tidspunkt vil produktet blive videreudviklet, således at det opretter

forbindelse til CapNordic's server, for at se om der er blevet oprettet nogle nye licens-filer, som den skal indlæse. Det er dog ikke en del af projektet, omkring den web-baserede licensmanager at få implementeret det i workflow produktet, da projektet kun handler om at få udviklet den web-baserede licensmanager.

- Den web-baserede licensmanager skal udvikles i Microsoft Visual Studio til web-udvikling og med VB.NET kode. Da der hos CapNordic A/S arbejdes i Microsofts udviklingssprog VB.NET, så skal den web-baserede licensmanager også udvikles i en blanding af HTML og VB.NET. På den måde, vil det være nemmere for andre medarbejdere hos CapNordic, at forstå hvordan den er lavet og evt. videreudvikle på den web-baserede licensmanager, hvis det på et senere tidspunkt skulle blive nødvendigt.

- Der udover har det været et krav fra CapNordic A/S' side, at det i første omgang var vigtigst at få hjemmesiden lavet, så den kunne lave de 3 licens-filer, med et simpelt design og hvor alle chefer og konsulenter hos CapNordic bare bruger det samme login. På et senere tidspunkt, som ikke er i forbindelse med bachelorprojektet, kan der så udvikles noget mere sikkerhed i hjemmesiden, så der blandt andet er seperat login for hver person i firmaet og få lavet noget mere sikkerhed så det er sværere for uvedkommende at opsnappe information. Senere kan udseendet også laves, så det bliver lidt mere spændende at se på.

1.2 Indledende tanker til projektet

Noget af det første jeg gik i gang med at overveje, var hvordan selve hjemmesiden skulle se ud og hvordan den skulle sættes op. Jeg ville helst gerne have, at hjemmesiden var så pæn og simpel, som overhovedet muligt og da CapNordic ikke havde nogle specifikke krav, til hvordan hjemmesiden skulle se ud, så tænkte jeg, at det for dem ville være bedst, hvis den lignede den gamle licensmanager i udseendet, så meget som overhovedet muligt. Dette ville gøre, at de vil have nemmere ved at gå til og bruge licensmanageren og den ville virke mere intuitiv at gå til. Jeg ville også gerne lave hjemmesiden, så det ville være nemt, at få et overblik over hvor man kunne få lavet de filer, man skulle give til kunden og så man i så få klik som muligt, kunne få genereret de filer der skal bruges. Før jeg overhovedet ville kunne gå i gang med at udvikle, så blev jeg også nødt til at finde ud af hvilken metode og hvilket udviklingsmiljø, jeg skulle bruge til at udvikle licensmanageren i. Det var, som nævnt, et krav at det skulle udvikles Microsoft Visual studio til web, men der findes flere versioner af Microsoft Visual Studio til web. Jeg valgte derfor at bruge den nyeste version, Microsoft Visual Studio 2012 til web, da jeg i den version ville kunne bruge de nyeste standarder, inden for web-udvikling (HTML5 og CSS3). Grunden til at jeg gerne ville kunne

bruge de nyeste standarder, var for at kunne udvikle en hjemmeside med den bedste mulige funktionalitet og så det ikke ville være nødvendigt at opdatere hjemmesiden alt for hurtigt. Efter at have bestemt mig for et udviklingsmiljø, så stod jeg over for et andet valg. I Microsofts udviklingsmiljø, er det nemlig muligt at udvikle hjemmesider på 3 forskellige måder:

- **Web pages:** Er den simpleste af Microsofts 3 metoder til udvikling af hjemmesider. I denne metode bruges der en blanding af HTML og Razor. Razor er et letvægts kodesprog, der kan skrives med enten C# eller Visual Basic kode. Det bliver brugt, til nemt at tilføje kode til en web-page som skal udføres på serveren inden selve siden bliver vist til brugeren. Ved at bruge Razor, er det også muligt at få adgang til en database.

- **MVC:** MVC er en 3-delt metode at udivkle hjemmesider på. M'et står for model, denne komponent bruges til at indeholde data og holde styr på det. V'et står for view og denne komponent bruges til at lave, det der grafisk skal vises til brugeren. C'et står for controller og denne komponent skal bruges til at kontrollere og holde styr på input fra data delen, til den visuelle del og omvendt, det er et bindeled mellem data og det som brugeren får vist. MVC metoden virker som noget der er godt at bruge, når der skal udvikles hjemmesider, hvor der vil være meget data der skal håndteres og hvis der vil være mange ændringer i data. Så hvis det er store avancerede hjemmesider der skal udvikles, så vil dette være en god metode at udvikle i.

- **Web forms:** Web forms bliver skrevet i en blanding af HTML og C# eller Visual Basic. I denne metode er det muligt at designe hjemmesider, ved brug af en Drag-n-Drop teknik, altså man kan trække en control, så som en knap eller en tekst boks ind på siden, hvor man vil have den skal være. Der udover er Web-forms en event-driven metode, det vil sige at man kan styre hvad der skal ske, når et bestemt event til en control forekommer. Denne metode vil kunne bruges til dynamiske hjemmesider hvor der skal bruges noget data fra en database for eksempel, men hvor der ikke er brug for, at have al for meget kontrol over det, da der ikke skal ske de store data beregninger.

Jeg valgte at udvikle den web-baserede licensmanager i Web-Forms metoden. Da det er en nogenlunde simpel hjemmeside der skal laves, med lidt data som der ikke skal laves de store beregninger på. Der udover synes jeg det ville være rart, at kunne sætte udseendet på de enkelte web-pages op ved hjælp af en designer og så også kunne styre hvad der skulle ske i de forskellige control-events.

2 CapNordic A/S

2.1 Om CapNordic A/S

CapNordic A/S er et mindre firma der ligger i Klampenborg, firmaet har hovedsageligt kunder i danmark, men har også nogle kunder i udlandet. Firmaet består af en gruppe konsulenter og en gruppe udviklere. Udviklerne er dem der laver selve produktet som CapNordic A/S sælger, de sidder og videreudvikler på produktet, så der kommer nye moduler og funktioner til produktet og prøver at forbedre nogle dele af produktet, for at se om det kan køre hurtigere og bedre, der udover fejlretter de også produktet, hvis der bliver indrapporteret en fejl i produktet. Konsulenterne er dem der tager ud til kunder og sætter produktet op og de kan lave små specielle ændringer, hvis et enkelt firma har brug for en ændring til en mindre del af produktet, som kun de skal bruge. Det er også konsulenterne der tager ud til nye mulige kunder, for at fortælle om produktet og sælge det. I og med at konsulenterne har kontakten til kunderne, så er det også dem firmaerne kommer til, hvis de vil tilkøbe et nyt modul, have lavet et nyt modul eller hvis de har opdaget en lille fejl. Konsulenterne tager så kontakt til udviklerne og giver dem besked om de ting firmaet godt vil have lavet.

2.2 CapNordic's produkt

Det produkt CapNordic A/S sælger og udvikler hedder "CapNordic Workflow", produktet gør det muligt at håndtere fakturaer elektronisk, i stedet for at de skal håndteres i hånden og ligge et fysisk sted, hvor de fylder. Alle dokumenterne ligger i en database, hvor al den nødvendige information omkring dokumenterne og hvad der skal bruges til de forskellige dokumenter ligger. Selve CapNordic's produkt er lavet så der er en hoveddel, som indeholder nogle standard komponenter og så er der flere moduler, som kunderne kan vælge at købe adgang til, alt efter hvad de har brug for. Standard produktet giver mulighed for at kunderne kan kontere deres faktura, holde styr på hvem der har kigget på en faktura og haft noget at gøre med den. De har også mulighed for at ændre hvem der har adgang til en eller flere bestemte fakturaer, der er mulighed for at kunne lave statistik over de fakturaer som firmaet har. Det er også muligt at styre brugerrettigheder, altså hvem der skal være administratorer og vil kunne oprette nye brugere, hvem der bare skal have lov til at kontere på fakturaer osv. For nogle kunder er det ikke nok bare at kunne bruge de ting som standard produktet giver adgang til, nogle har måske brug for at kunne lave en rapport over en faktura, hvor de kan få printet noget bestemt information ud omkring et dokument, nogle har måske brug for at kunne holde styr på, hvornår forskellige brugere er på ferie så hvis der skal konteres en faktura så kan det blive videresendt til en anden imens. Sådanne module vil være tilgængelige til de kunder der vælger at tilkøbe de moduler. Derfor er produktet ikke helt ens ude hos alle

kunder, da forskellige kunder har forskellige behov og derfor har de valgt at lave produktet modul-baseret, så hver kunde kan få det de har brug for og ikke skal betale for ting de ikke kommer til at bruge.

2.2.1 Workflow V5

Workflow Version 5 er CapNordic's gamle produkt, det er udviklet i VB6 og har været det produkt de har arbejdet med, indtil for 1-2 år siden. Der er stadig en del kunder der bruger det gamle produkt og firmaer har stadig mulighed for at købe dette produkt og nye moduler til dette produkt. Denne version bliver der dog ikke udviklet nye moduler til, så vil en kunde have adgang til nogle af de nyere moduler, så skal de købe den nye version.

2.2.2 Workflow V8

Workflow version 8 er CapNordic's nye produkt som er det der udvikles på nu og det udvikles i sproget VB.NET. Det nye og det gamle produkt er sådan set ens, da de begge gør at firmaer kan håndtere deres fakturaer og dokumenter elektronisk. Forskellen på den nye og den gamle version er at en del af funktionaliteten er forbedret og kører hurtigere end i version 5. Udseendet er også blevet ændret en hel del, så det er en hel del pænere at se på. Når der skal udvikles et nyt modul, så vil det kun være tilgængeligt til den nye version. Workflow version 8 er begyndt at blive solgt til nyere kunder og blive sat op hos dem, plus nogle kunder der har haft det gamle produkt har valgt at købe den nye version i stedet.

2.3 Licensfiler

CapNordic A/S gør brug af 3 typer licens-filer, til at holde styr på kundernes produkter og for at kontrollere at de enkelte kunder kun får det de har betalt for og dermed har rettighed til at bruge. Alle filerne er krypterede og det information de gemmer er også krypteret, for at sikre at der ikke bliver lavet licens-filer ulovligt af folk som ikke må lave licens-filer.

2.3.1 Hovedlicens/Kundelicens fil

Når en kunde køber et af CapNordic's produkter, Workflow version 5 eller Workflow version 8, så får de en Hovedlicens/kundelicens med til produktet. Denne fil gør at produktet kan bruges og det angiver hvilket firma der bruger produktet. Meningen med denne licens-fil er også at kontrollere, at en kunde ikke modtager en Vedligeholdelses licens fil eller en modul-licens fil, fra en anden og på den måde kan blive ved med at bruge produktet og få nye moduler uden at betale for det.

2.3.2 Vedligeholdelses fil

Når en kunde køber Workflow version 8 eller version 5, så har de ret til at bruge produktet i et år. Når det år er gået, så skal de betale for at få fornyet produktet, så de fortsat kan blive ved med at bruge produktet. For at kontrollere at kunderne betaler når produktet skal fornyes, så laver CapNordic A/S en Vedligeholdelses fil. Denne fil angiver hvornår produktet til en bestemt kunde skal udløbe. Så når produktet er udløbet eller er ved at udløbe, så kan der indlæses en Vedligeholdelses licens-fil i produktet, hvorefter der vil blive sat en ny dato for hvornår produktet skal fornyes. Et firma får kun en ny vedligeholdelses licens-fil, når de har betalt for det, så på den måde kan CapNordic A/S sikre sig at kunderne betaler for at fornye deres produkt og ikke bare bruger det uden at betale.

2.3.3 Modul fil

Som tidligere nævnt er Workflow version 8 og version 5 et modul baseret produkt. Derfor skal der holdes styr på at en kunde kun kan bruge de moduler der er betalt for og at de ikke får adgang til moduler, de ikke har ret til at bruge. Der udover skal det være muligt at låse op for moduler hos en bestemt kunde, når de har betalt for et eller flere moduler. En modul-licens fil indeholder derfor information omkring hvilke moduler der skal gives adgang til, hvor mange brugere en kunde må have eller hvor mange flere de vil have, hvor mange dokumenter de har plads til eller hvor mange flere dokumenter de vil have plads til osv. Modul-licens filen skal indlæses i produktet af en kunde, når de har tilkøbt noget mere til deres produkt. Når filen så bliver læst ind, så vil de få adgang og rettighed til at bruge det de nu har købt.

3 Udvikling

3.1 Den web-baserede licensmanager

Selve licensmanageren består af 2 dele, den ene er de web-sider der bliver vist for brugeren og hvor brugeren kan vælge hvilke licenser der skal laves, til hvilken kunde licenserne skal laves og hvilken information der skal bruges til at lave licens-filerne. Den anden del er de Visual Basic klasser, der kan oprette forbindelse til databasen og hente information op eller ligge det ned i databasen og generere de licens-filer der skal laves.

3.1.1 Indhold i web-delen

Selve web-delen består af 6 web-sider, som giver brugeren en grænseflade at arbejde med, når der skal oprettes licens-filer eller når brugeren vil se noget information omkring en kunde og diverse andre ting der har med licensmanageren at gøre. Der udover består web-delen også af en .master fil og en .css fil.

- **Web-side 1, Login.ASPX:** Login siden er den første man kommer til som bruger, når man vil bruge licensmanageren. Meningen med denne side, er at give brugeren mulighed for at indtaste brugernavn og kodeord til databasen, for at se om brugeren rent faktisk har adgang til at bruge licensmanageren. Bruger man det forkerte brugernavn og kodeord, så vil der ikke ske noget, men bruger man det rigtige, så vil man blive logget ind og dermed få adgang til de funktioner der er i licensmanageren. Når en bruger har logget korrekt ind, vil der blive gemt en variabel i den session brugeren har, så de andre web-sider brugeren kan komme ind på, ved at han/hun har adgang til de sider og brugeren vil blive sendt hen til den web-side der hedder *"Frontpage.ASPX"*. Prøver en bruger gennem adresse baren at komme ind på en af de andre web-sider uden at have logget ind, så vil brugeren blive sendt tilbage til login siden for at logge ind.

- **Web-side 2, Frontpage.ASPX:** Denne side er forsiden, det er som nævnt den første side en bruger kommer ind på, når man har logget ind. På denne side er det muligt at se al information, der er om en kunde. Det vil sige, at man kan se de generelle ting, så som navnet på firmaet, addressen, telefon nummer osv. Man kan også se hvilken version af produktet en kunde har, hvem hos kunden der er kontakt person og hvem fra CapNordic, der er konsulent ude hos kunden. Der udover er det muligt for en bruger at slette, oprette og ændre information for en kunde.

- **Web-side 3, License.ASPX:** På denne side er det muligt at oprette en hovedlicens fil, ved at vælge en kunde og om det skal være en

licensfil til Workflow version 8 eller Workflow version 5.

- **Web-side 4, Maintenance.ASPX:** På denne side er det muligt at oprette en vedligeholdelseslicens fil. Dette kan gøres ved at vælge en kunde, vælge en ny dag hvor produktet skal udløbe og om det er en fil der skal bruges til Workflow version 5 eller Workflow version 8.

- **Web-side 5, Modules.ASPX:** På denne side er det muligt for brugeren at oprette de modul licens-filer, der skal til for at åbne op for nye moduler osv. Her kan en bruger vælge hvilken kunde der skal have filen, hvilke nye moduler de skal have adgang til, hvor mange brugere, dokumenter, indbakker og statistikker de skal have. Her skal der selvfølgelig også vælges, om det er den nye eller gamle workflow version som filen skal bruges til.

- **Web-side 6, Download.ASPX:** På denne side er det muligt for brugeren at downloade en licens fil. Så når en bruger har oprettet en ny licens fil, så kan brugeren også hente denne fil ned lokalt, til den computer han/hun sidder ved.

- **.Master filen:** .Master-filen bliver brugt til at lave et generelt udseende, som skal bruges på flere web-sider, dette gør at man ikke skal bruge tid på at lave et overordnet udseende, på flere sider og sørge for at de er helt ens. Der skal det bare specificeres på de enkelte web-sider, at de skal bruge udseendet fra .Master-filen og så er man sikret at de har det samme overordnet udseende. Det gør også at man ikke skal ind flere steder og ændre noget hvis der skal ændres noget. Siderne der gør brug af denne .Master fil til at have det samme overordnede udseende er: "*Fronpage.ASPX*", "*License.ASPX*", "*Maintenance.ASPX*", "*Modules.ASPX*" og "*Download.ASPX*".

- **.CSS filen:** .CSS filen bliver brugt til at specificere hvordan forskellige controls, så som knapper, tekst bokse, labels osv. skal se ud, hvis man vil have at de skal have et anderledes udseende, end det de normalt er sat til. Det gør også, at man nemt kan sørge for, at flere tekst bokse ser ud på samme måde, uden at man skal specificere det på hver tekst boks.

3.1.2 Visual Basic klasser

- **Connection.VB:** Denne klasse bliver brugt til at oprette en forbindelse til databasen, det vil sige, hvis der skal hentes noget op fra databasen eller skal der gemmes noget i databasen, så skal der oprettes et objekt af denne klasse, for at kunne få åbnet forbindelse til databasen.

- **DbFunctions.VB:** Denne klasse bliver brugt når, der skal hentes noget op fra databasen og når der skal gemmes noget ned i databasen.

Det er i denne klasse, at alle de metoder som web-siderne bruger, til at hente og gemme data i databasen ligger.

- **CreateFile.VB:** Denne klasse bliver brugt når der skal genereres en licens-fil, det er altså her at metoderne til at oprette en kundelicens-fil, vedligeholdelses-licens fil og modul-licens fil ligger. Det er også i denne klasse, at filerne bliver krypteret, så de ikke kan kopieres eller genereres ulovligt.

3.2 Udvikling af Visual Basic klasser

3.2.1 Connection.VB

"*Connection.VB*" klassen indeholder 4 variable, som indeholder information omkring, hvor databasen er, hvilken port man skal bruge, for at få adgang til databasen, hvad databasen hedder og en variabel der kan åbne og lukke for adgangen til databasen. Der udover er der en property, der kan returnere den variabel, der kan åbne og lukke for forbindelsen til databasen og så er der en metode, til at teste et login. Følgende kode udsnit fra source koden, viser hvad der sker i test metoden:

```
Public Function testLogin(ByVal user As String, ByVal password As
    String) As Boolean
    Dim conStr As String = "DRIVER={SQL Server};Server=" & host &
        "," & port & ";Database=" & db & ";UID=" & user & ";PWD=" &
        password & ";"

    Try
        Dim testCon As Odbc.OdbcConnection = New
            Odbc.OdbcConnection(conStr)

        testCon.Open()
        testCon.Close()

        _objCon = testCon
        Return True
    Catch ex As Exception
        Return False
    End Try
End Function
```

Metoden modtager et brugernavn og kodeord, når en bruger prøver at logge ind. Der bliver så lavet en connection streng, med det modtagede brugernavn og kodeord. Dernæst bliver der oprettet et ODBC connection objekt, med connection strengen og der prøves så at åbne og lukke en forbindelse til databasen. Grunden til at det sker i en try-catch blok, er hvis der ikke kan skabes forbindelse til databasen, med det brugernavn

og kodeord der er blevet modtaget, så vil der blive kastet en exception og så vil metoden returnere "*False*". Kan der derimod successfuldt oprettes forbindelse til databasen, så vil ODBC connection objektet blive gemt i den lokale variabel, som skal gøre det muligt at åbne og lukke for forbindelsen til databasen og metoden vil så returnere "*True*".

3.2.2 DbFunctions.VB

Denne klasse består af 2 metoder, en metode til at hente ting op fra databasen "*getDataTable*" og en metode "*NonQuery*", til at gemme eller ændre information nede i databasen. Følgende kode udsnit, viser hvad der sker i metoden "*getDataTable*":

```
Public Function getDataTable(ByVal conn As Connection, ByVal
    sqlQuery As String) As DataTable
    Dim dt As DataTable = New DataTable
    Dim objCon As Odbc.OdbcConnection = conn.objCon
    Dim objCommand As Odbc.OdbcCommand = New Odbc.OdbcCommand

    objCommand.CommandText = sqlQuery

    objCon.Open()

    objCommand.Connection = objCon

    dt.BeginLoadData()

    Using adapter As New Odbc.OdbcDataAdapter(objCommand)
        adapter.Fill(dt)
    End Using

    dt.EndLoadData()

    objCon.Close()
    Return dt
End Function
```

Metoden modtager variabel af typen "*Connection.VB*" og en streng der indeholder en sql-query. Der oprettes så en datatabel, som kommer til at indeholde det der bliver hentet op fra databasen, ud fra hvad der står i sql strengen. Den ODBC connection variabel der kan åbne og lukke for forbindelsen til databasen, bliver hentet ud af "*Connection.VB*" objektet og der bliver oprettet et ODBC command objekt. Dette objekt bruges til at sende sql query'en ned til databasen, gennem den forbindelse ODBC connection variablen har åbnet. Datatabellen sættes så i en tilstand, så den er klar til at modtage den information som ODBC command objektet får tilbage fra databasen, der bruges så også en adapter til at ligge det

ind i databasen. Når det er sket, sættes datatabellen tilbage til en normal tilstand, så man kan hente ting ud af datatabellen, forbindelsen til databasen lukkes og datatabellen bliver returneret. Følgende udsnit viser hvad der sker i metoden "*NonQuery*":

```
Public Sub NonQuery(ByVal conn As Connection, ByVal sqlQuery As
    String)
    Dim objCon As Odbc.OdbcConnection = conn.ObjCon
    Dim objCommand As Odbc.OdbcCommand = New Odbc.OdbcCommand
    objCommand.CommandText = sqlQuery

    objCon.Open()

    objCommand.Connection = objCon

    objCommand.ExecuteNonQuery()

    objCon.Close()
End Sub
```

Ovenstående metode modtager ligesom den forrige metode, et objekt af typen "*Connection.VB*" og en streng, der indeholder en sql query, der angiver hvad der skal gemmes eller ændres i databasen. Forskellen mellem den her metode og den forrige, er at der i denne ikke er brug for nogen datatabel eller adapter, til at hente ting op fra databasen, i stedet for bliver der udført en "*ExecuteNonQuery*" kommando i ODBC command objektet, som sender sql strengen ned i databasen og udfører den uden at få noget data returneret. Udover det og at metoden ikke returnerer noget, så sker der det samme i de 2 metoder.

3.2.3 CreateFile.VB

Klassen "*CreateFile.VB*" indeholder 5 metoder, en metode "*Crypto*", som bruges til at kryptere noget tekst, en metode "*random2Chars*", som bruges til at lave 2 fuldstændigt tilfældige karakterer og så er der de 3 metoder "*createLicenseFile*", "*createMaintenanceFile*" og "*createModuleFile*", som bruges til at generere de 3 licensfiler, henholdsvis hovedlicens filen, vedligeholdelseslicens filen og modul licens filen. Følgende udsnit er fra metoden "*crypto*":

```
Public Function Crypto(ByVal InString As String, ByVal PrivateKey
    As String, ByVal PublicKey As String) As String
```

Som det ses, så modtager "*Crypto*" 3 strenge, "*InString*" som er den tekst der skal krypteres, "*PrivateKey*" som indeholder den hemmelige nøgle, der skal bruges til at kryptere teksten og så er der "*PublicKey*" som er den offentlige nøgle, der skal bruges i krypteringen. Metoden returnerer

en streng, som er den krypterede version af teksten, der blev modtaget sammen med den offentlige og hemmelige nøgle. Det første metoden gør, er at konvertere den hemmelige nøgle og den offentlige nøgle fra en streng, om til et byte array. Følgende kode udsnit viser hvordan det gøres:

```
For i = 1 To Len(PrivateKey)
    KeyList(i) = CByte(Asc(Mid(PrivateKey, i, 1)))
Next i
```

For-løkken kører igennem hele den hemmelige nøgle og hver karakter bliver lavet om til den tilsvarende byte-værdi, som bliver gemt i et array. Det foregår på samme måde med den offentlige nøgle, den bliver bare gemt i et andet array. Dernæst går metoden ind i endnu en for-løkke, som løber igennem hele den tekst, der skal krypteres. Hver karakter bliver så krypteret ved hjælp af den hemmelige nøgle og den offentlige nøgle. Følgende kode viser hvordan det sker:

```
newChar = Chr((Asc(Mid(InString, i, 1)) Xor KeyList(j)) Xor
    PubList(k))
myC.Append(newChar)
```

Det første der sker her, er at en karakter fra teksten, bliver konverteret om til den tilsvarende byte-værdi. Denne værdi bliver så XOR'et med en byte-værdi fra den hemmelige nøgle, så der bliver dannet en ny byte-værdi. Den nye byte-værdi bliver så XOR'et med en byte-værdi fra den offentlige nøgle, så der igen bliver dannet en ny byte-værdi. Den sidste byte-værdi bliver så konverteret om til den karakter, der svarer til den værdi der er blevet dannet. Til sidst bliver den nye karakter tilføjet til "*myC*", som er en StringBuilder. Når for-løkken er nået igennem hele teksten og alle karakterer er blevet krypteret og tilføjet til StringBuilderen, så vil den krypterede tekst der er dannet i StringBuilderen blive returneret.

Metoden "*random2Chars*" er blevet lavet, fordi det var nødvendigt at tilføje 2 ekstra karakterer til teksten, for at sørge for at den havde en korrekt længde. Grunden til at det var nødvendigt, er fordi der, før Workflow version 8 blev lavet, var en ældre licensmanager end den der bliver brugt nu. Den licensmanager var udviklet i VB6 og på grund af den måde den encodede filer på når de blev lavet, så ville den krypterede tekst være 2 karakterer længere. De 2 ekstra karakterer bliver skåret væk, i både Workflow version 5 og version 8, så for at der ikke bliver skåret noget af den krypterede streng, der bliver dannet i den nye licensmanager, som ikke må skæres væk, så tilføjes der altså 2 tilfældige karakterer, der kan fjernes uden at det gør noget ved teksten. Følgende kode udsnit er hvad der sker i metoden "*random2Chars*":

```
Private Function random2Chars() As String
    Dim randomString As String = ""
    Dim randomGenerator As New Random()
```

```
    Dim bytes(1) As Byte
    bytes.SetValue(Byte.Parse(randomGenerator.Next(0, 127) & ""), 0)
    bytes.SetValue(Byte.Parse(randomGenerator.Next(0, 127) & ""), 1)

    randomString = System.Text.Encoding.UTF8.GetString(bytes)

    Return randomString
End Function
```

Der bliver oprettet en streng som skal indeholde de 2 tilfældige karakterer, en random generator, som kan generere tilfældige tal og der bliver lavet et byte array, der kan indeholde 2 byte-værdier. På hver af de 2 pladser i byte arrayet, bliver der dannet 2 tilfældige tal, i intervallet 0 til 127 og disse 2 tal bliver konverteret om til en byte-værdi. Grunden til at det er i intervallet 0 til 127 er for at de karakterer der bliver dannet er en af de 128 mulige karakterer der i en ASCII encoding. Byte arrayet med de 2 byte-værdier, bliver så lavet om til en streng med 2 karakterer og den bliver så returneret.

Følgende kode er et udsnit fra metoden "*createLicenseFile*":

```
Public Function createLicenseFile(ByVal licName As String, ByVal
    version As String, ByRef _filePath As String) As Boolean
```

Her kan det ses at metoden modtager en streng "*licName*", som er licensnavnet på den kunde, som hovedlicens filen skal laves til. Der bliver modtaget en streng "*version*", som angiver om licens filen skal laves til Workflow version 5 eller Workflow version 8. Den sidste ting den modtager er en reference til en streng, der indeholder lokationen på serveren, hvor filen bliver gemt. Metoden returnerer en boolean værdi, alt efter hvordan det går med at få lavet hovedlicens filen. Det vil sige at hvis der går noget galt undervejs når teksten bliver krypteret eller når filen bliver gemt, så vil den returnere "*False*". Går alt derimod godt, jamen så returnerer den "*True*".

Metoden sender licensnavnet ned til "*Crypto*" metoden, sammen med en hemmelig nøgle og en offentlig nøgle, så licensnavnet kan blive krypteret. Det næste der så skal ske er at den krypterede tekst, skal laves om til et byte array, men det skal konverteres ved at bruge en bestemt encoding, alt efter om det er Workflow version 5 eller Workflow version 8, da de jo som tidligere nævnt indlæser filer forskelligt. Følgende udsnit viser hvilken encoding der skal bruges når licens filen skal bruges til Workflow version 5:

```
cryptoBytes = Encoding.Default.GetBytes(cryptoText)
```

Her bliver den krypterede tekst omdannet til et byte array, ved at bruge en default encoding. Denne encoding er en Single-Byte Character-set (SBCS), det vil sige at hver karakter i det default sæt af karakterer som encodingen har, har hver deres byte. Andre SBCS-encodings der ikke er default, vil

have et andet karakter sæt og med andre byte-værdier. Det næste udsnit af source koden, viser hvilken encoding der bliver brugt til de licens filer der skal bruges i Workflow version 8:

```
cryptoBytes = New UTF8Encoding(True).GetBytes(cryptoText)
```

Her bliver der brugt en UTF-8 encoding, dette er en Unicode-encoding, hvilket vil sige at hvert karakter, vil have den samme byte-værdi uanset hvilken Unicode encoding der bliver brugt. Hvis man prøver at konvertere fra en Unicode-encoding til en SBCS-encoding eller omvendt, så vil der enten forekomme korruption af data eller der vil blive mistet data, hvilket er grunden til at der ikke kan bruges samme slags encoding til begge versioner.

Når den krypterede tekst er blevet lavet om til et byte array, så oprettes der en "*FileStream*". Denne "*FileStream*" bruges til at oprette en fil et sted på serveren og skrive byte arrayet ud til filen, så den kommer til at indeholde det der er blevet krypteret. I følgende kode udsnit ses det hvordan det foregår:

```
Dim fs As FileStream = New FileStream(filePath, FileMode.Create)
fs.Write(cryptoBytes, 0, cryptoBytes.Length)
fs.Close()
```

Først bliver "*FileStream'en*" oprettet med en streng "*filePath*", som indeholder den lokation hvor filen skal gemmes og det bliver angivet at "*FileStream'en*" skal bruges til at oprette en fil. Dernæst bliver alle bytes i byte arrayet, sendt til "*FileStream'en*" og den skriver så alt ind i filen. Til sidst skal "*FileStream'en*" lukkes så filen kan bruges. Alt det ovenstående omkring kryptering og generering af licens filen, sker i en try-catch. Grunden til dette, er for at fange de fejl der kan ske og hvis der sker en fejl så sørge for at metoden returnerer "*False*". Er der ikke sket nogle fejl, så vil den reference der er modtaget, blive sat til at indholde lokationen for licens filen og metoden vil returnere "*True*"

Følgende kode er taget fra metoden "*createMaintenanceFile*":

```
Public Function createMaintenanceFile(ByVal licName As String,
    ByVal maintenanceDate As Date, ByVal version As String, ByRef
    _filePath As String) As Boolean
```

Metoden "*createMaintenanceFile*" modtager en streng "*licName*", som er licensnavnet til den kunde, som vedligeholdelseslicens filen skal laves til, der bliver modtaget en dato "*maintenanceDate*", som er den nye dato for hvornår produktet skal fornyes, der bliver også modtaget en streng "*version*", som angiver den version, som filen skal bruges til og der er også modtaget en reference "*_filePath*" til en streng, der skal indeholde lokationen på serveren hvor filen er gemt. Denne metode returnerer også en boolean der indikerer om filen er blevet lavet eller ej. Datoen for den nye fornyelse og

licensnavnet bliver sendt ned til metoden "*Crypto*", sammen med nøglerne for at få en krypteret tekst. Når den krypterede tekst er lavet, så foregår det på samme måde som i metoden "*createLicenseFile*", med at den krypterede tekst bliver konverteret til et byte array, hvor der bliver brugt "*Default SBCS encoding*" hvis det er til Workflow version 5 eller "*UTF-8 encoding*", hvis vedligeholdelseslicens filen skal bruges til Workflow version 8. Der efter bliver der lavet en "*FileStream*", som opretter en fil på en given lokation og får gemt al det krypterede data ned i filen. Går alt dette godt, bliver lokations referencen "*_filePath*" sat til at indeholde lokationen på filen og metoden vil returnere "*True*". Går noget galt i forløbet, så vil referencen ikke blive sat og metoden vil returnere "*False*".

Følgende udsnit fra source koden er fra metoden "*createModuleFile*":

```
Public Function createModuleFile(ByVal licName As String, ByVal
    modules As ArrayList, ByVal licUsers As Integer, ByVal licDocs
    As Integer, ByVal licInboxes As Integer, ByVal statViews As
    Integer, ByVal add As Boolean, ByVal version As String, ByRef
    _filePath As String) As Boolean
```

Denne metode modtager ligesom "*createLicenseFile*" og "*createMaintenanceFile*" et licensnavn til kunden "*licName*", en streng der angiver hvilken version filen skal laves til "*version*" og en reference til det sted filen er gemt "*_filePath*". Der udover så modtager denne metode også en ArrayList "*modules*", som er en liste over de moduler kunden har købt adgang til, en integer "*licUsers*" der angiver antal brugere, "*licDocs*" der angiver antal dokumenter, "*licInboxes*" der angiver antal indbakker, "*statViews*" der angiver antal statistikker og en boolean værdi "*add*" som angiver om de værdier der er angivet af "*licUsers*", "*licDocs*", "*licInboxes*" og "*statViews*" er noget der skal ligges til den nuværende værdi hos kunden eller om det er den totale værdi som kunden må have. Metoden returnerer også en boolean værdi der fortæller om filen er blevet lavet eller om der skete en fejl.

Metoden "*createModuleFile*" kører på samme måde som "*createLicenseFile*" og "*createMaintenanceFile*", den eneste forskel, er dog bare at i de 2 andre metoder, der er den tekst der skal krypteres allerede lavet, men i "*createModuleFile*" der skal teksten laves ud fra det der er modtaget i metoden. Det bliver gjort ved at tilføje noget tekst til en StringBuilder "*moduleStr*", følgende kode er et eksempel på hvordan det foregår:

```
If add = True Then
    moduleStr.Append("#ModuleLicensVersion2-add#" & "#NewRecord#")
Else
    moduleStr.Append("#ModuleLicensVersion1#" & "#NewRecord#")
End If
```

Her bliver der lavet noget tekst, der angiver hvilken metode der skal bruges, når filen bliver læst ind, så produktet ved om værdierne fra "*li-*

cUsers", "*licDocs*", "*licInboxes*" og "*statViews*" skal ligges til de nuværende værdier eller om det skal være de totale værdier. Der bliver også lavet noget tekst, så når filen bliver læst ind, så ved produktet hvor mange brugere der skal tilføjes eller være:

```
If licUsers > 0 Then
    moduleStr.Append("#FieldStart#Licens_users#FieldEnd##FieldValueStart#"
        & licUsers & "#FieldValueEnd##NewRecord#")
End If
```

Som det kan ses, så bliver der kun tilføjet en streng til StringBuilderen, hvis værdien er større end 0. Det er for at sikre, at værdien hos kunden kun bliver ændret, hvis den skal ændres, ellers kunne det være en risiko, at den totale værdi hos kunden blev sat til 0, hvis parametren "*add*" angiver at værdierne for "*licUsers*", "*licDocs*", "*licInboxes*" og "*statViews*" skal være de totale værdier. Der bliver lavet noget tekst for "*licDocs*", "*licInboxes*" og "*statViews*" på lignende måde.

Der bliver ligeledes lavet noget tekst, for hver af de moduler kunden skal have adgang til så når filen bliver læst ind, så vil moduler blive gjort til rådighed. Det sker på følgende måde:

```
For Each _module As String In modules
    moduleStr.Append("#FieldStart#" & _module &
        "#FieldEnd##FieldValueStart#" & "TRUE" &
        "#FieldValueEnd##NewRecord#")
Next
```

Som det kan ses, så er der lavet en for-løkke, som løber igennem hele ArrayListen med de moduler der skal åbnes op for og der bliver tilføjet noget tekst til StringBuilderen for hvert af modulerne. Når hele teksten er blevet lavet, så bliver den sendt ned til metoden "*Crypto*", sammen med en hemmelig og offentlig nøgle. Den krypterede tekst bliver sendt til en FileStream som gemmer det ned i en fil. Går alt som det skal, så bliver lokations referencen sat til det sted hvor filen er gemt og metoden returnerer "*True*" og ellers returnerer metoden "*False*".

3.2.4 Fejl håndtering i Visual Basic klasserne

For at håndtere eventuelle fejl der skulle forekomme og som kan få det hele til at fejl, så er der lavet exception håndtering over det hele. Dette er gjort for at man ikke nødvendigvis som bruger behøver at starte helt forfra, hvis der skulle ske en fejl, men at man kan få et forsøg igen til at foretage hvad man nu er i gang med. Der udover så sikre det også at hele hjemmesiden ikke crasher.

3.3 Udvikling af web-delen

3.3.1 Site.Master

Site.Master siden som indeholder det overordnede udseende for "*Frontpage.ASPX*", "*License.ASPX*", "*Maintenance.ASPX*", "*Module.ASPX*" og "*Download.ASPX*", består af 3 ting. En hovedtitel:

```
<h1>CapNordic A/S Licensmanager</h1>
```

Der er en uordnet liste:

```
<ul id="menu">
   <li><a href="Frontpage.aspx">Frontpage</a></li>
   <li><a href="License.aspx">License file</a></li>
   <li><a href="Maintenance.aspx">Maintenance file</a></li>
   <li><a href="Module.aspx">Module file</a></li>
</ul>
```

I denne liste er hvert element et link til hver af de sider, som brugeren har lov til gå frit imellem når han/hun er logget ind. Der udover er der også en "*ContentPlaceHolder*":

```
<asp:ContentPlaceHolder id="MainContent" runat="server">

</asp:ContentPlaceHolder>
```

Det er i denne "*ContentPlaceHolder*" at alle de controls, der er i "*Frontpage.ASPX*", "*License.ASPX*", "*Maintenance.ASPX*", "*Module.ASPX*" og "*Download.ASPX*" skal være. Den har fået et id ("*MainContent*") så siderne kan tilføje deres controls til, "*ContentPlaceHolderen*" når en bruger klikker sig hen på en af siderne.

3.3.2 Style1.CSS

Denne CSS fil bliver brugt af "*Site.Master*", til at definere hvordan det overordnede udseende skal se ud. Man kan sige at "*Site.Master*" siden, bliver brugt til at angive hvilke ting der skal være i det overordnede udseende og "*Style1.CSS*" bliver brugt til, at angive hvordan tingene i "*Site.Master*" skal se ud. Følgende kode udsnit fra "*Style1.CSS*" viser, hvordan kroppen af de forskellige web-sider skal være, altså hvor store siderne skal være og hvilken farve baggrunden skal have.

```
body {
    background-color: #C2D9F7;
    height: 485px;
    width: 1172px;
}
```

Hex-værdien "*#C2D9F7*" angiver at baggrunds farven er en lyseblå farve. Denne farve er blevet valgt da det er en farve der bliver brugt utroligt meget i CapNordic A/S's produkt.

Figure 1: Baggrundsfarven #C2D9F7.

Det næste der bliver defineret i CSS filen, er hvordan titelen på hjemmesiden skal være, det kan ses på følgende kode:

```
h1 {
    color: #000099;
    width: 1170px;
}
```

Her kan det ses hvilken farve teksten på titelen skal have, hex-værdien "*#000099*" angiver at det skal være en mørkeblå farve. Det er også bestemt at bredden på titelen, skal fylde næsten det hele af bredden af hjemmesiden.

Figure 2: Billede af Titel på licensmanageren.

Det sidste i CSS filen, er hvordan selve den uordnede liste skal se ud for brugeren. Det er blevet defineret af forskellige dele, fordi der er flere komponenter i en uordnet liste. Følgende kode viser hvordan selve menu'en overordnet skal være og positionen af den:

```
ul#menu {
    padding: 0px;
    position: relative;
    margin: 0;
    top: 0px;
    left: 0px;
    width: 934px;
}
```

Her er det angivet, at der ikke skal være noget mellemrum mellem listen og det der er til venstre for listen, det er bestemt af at padding er sat til 0 pixels. Postitionen er sat til at være relativ, så listens plads er defineret ud fra de ting der er omkring den. Margin er sat til 0, så de elementer der er i listen ikke har noget tom plads mellem sig og kanten af listen. Der er også specificeret en bredde på listen. Alle de elemeneter der er i listen er sat til

at blive vist på en horisontal linie, i stedet for at have en vertikal liste af punkter, det kan ses på følgende kode udsnit:

```
ul#menu li {
    display: inline;
}
```

Da elementerne i listen bare er nogle links til de andre web-sider, så er det blevet lavet så hvert element minder mere om en knap i stedet for bare et link, det er gjort med følgende udsnit fra "*Style1.CSS*":

```
ul#menu li a {
background-color: #ffffff;
padding: 10px 20px;
text-decoration: none;
line-height: 2.8em;
color: #034af3;
}
```

Baggrundsfarven til elementerne i listen, er sat til at være hvid ved at bruge hex-værdien "*#ffffff*". Der udover er der sat en padding på 10 gange 20 pixels omkring hvert link, så det sted man kan trykke på er 10 pixels højere, både over og under linket og 20 pixels bredere på hver side af linket. Den tekst der er skrevet i hvert element, har fået en mørkeblå farve, som det er angivet med hex-værdien "*#034af3*". Det sidste der er blevet defineret, er at baggrundsfarven til elementerne i listen, ændrer sig til en grøn farve når man holder musen over et elements område. Det viser følgende kode:

```
ul#menu li a:hover {
background-color: #24e822;
}
```

Figure 3: Uordnet liste på Master.Site.

3.3.3 Login.ASPX

Login siden er rigtig simpel, de eneste controls der er på denne web-side, er 2 labels, 2 tekst-bokse og en knap. Brugeren skal i den ene tekst-boks indtaste brugernavn og i den anden skal der indtastes kodeord. Følgende kode udsnit viser hvordan tekst-boksen til at indtaste kodeord er lavet:

```
<asp:TextBox ID="txtBoxPassword" runat="server"
    AutoCompleteType="Disabled" BorderStyle="Solid"
    TextMode="Password"></asp:TextBox>
```

Som det ses, så er "*TextMode*" sat til at være af typen "*Password*". Ved at gøre dette, så bliver de tegn der er indtastet i tekst-boksen, vist som prikker. På denne måde er det sikret at folk ikke kan se hvad kodeordet er når det bliver indtastet i tekst-boksen. Tekst-boksen til brugernavn minder meget om den til kodeord, den eneste forskel er at tekst-boksen til at indaste brugernavn ikke har fået defineret nogen type til "*TextMode*", så i denne tekst-boks vil teksten blive vist normalt.

Når en bruger har indtastet et brugernavn og kodeord og gerne vil logge ind, så skal de trykke på knappen. Følgende kode udsnit viser hvordan knappen er lavet:

```
<asp:Button ID="btnLogin" runat="server" Text="Login"
    OnClick="login"/>
```

Når en bruger klikker på knappen, så skal der helst ske noget bestemt, for at dette sker, så er det blevet defineret hvilken metode der skal udføres når control-eventet "*OnClick*" forekommer. I dette tilfælde, er det blevet specificeret at metoden "*login*" skal køres, når brugeren trykker på knappen. Følgende udsnit af source koden, viser hvad der sker i metoden "*login*" når knappen er blevet klikket på:

```
Sub login()
    Dim username As String = txtBoxUsername.Text
    Dim password As String = txtBoxPassword.Text

    Dim connection As Connection = New Connection

    If connection.testLogin(username, password) = True Then
        Session(connSessionVarName) = connection

        Response.Redirect("Frontpage.aspx")
    End If
End Sub
```

I denne metode bliver det tekst, der er indtastet i de 2 tekst-bokse hentet og gemt i 2 variable. Der bliver så oprettet en "*Connection*"-variabel, som bruges til at teste om det brugernavn og kodeord der er indtastet, er et gyldigt login til licensmanageren. Hvis det er gyldigt, så vil den "*connection*" variabel blive gemt i en sessions variabel, så hjemmesiden ved, at brugeren har adgang til at se de andre web-sider og brugeren vil så blive sendt hen til forsiden. Er det derimod ikke et gyldigt login, så vil der ikke ske noget.

3.3.4 Frontpage.ASPX

Forsiden er delt op i 2 dele, en højre og en venstre side. I den venstre side er der en boks som indeholder en liste af alle de kunder, som CapNordic har. Følgende kode viser hvordan den control er sat op:

```
<asp:ListBox ID="listBoxCustomer" runat="server" Font-Size="Small"
    Height="328px" Width="230px" AutoPostBack="true"
    OnSelectedIndexChanged="listBoxCustomer_SelectedIndexChanged">
<asp:ListItem>[---- Vaelg en kunde ----]</asp:ListItem>
```

Denne boks har fået et ID lige som alle andre controls, i de web-sider der er i licensmanageren. Dette ID gør det muligt at få fat i denne boks control, så man kan hente data eller ligge data ind i control'en, fra noget server kode. Det er også specificeret hvor stor boksen skal være og tekst-størrelsen på den tekst der er i boksen. Der udover er boks controllerens property "*AutoPostBack*" sat til at være "*True*", dette er gjort så hver gang der sker en ændring i boksen, så fortager siden et "*PostBack*", der gør at siden bliver kørt igen. På den måde er det muligt at få hentet det data der skal hentes op omkring en kunde, når der bliver valgt en kunde fra listen, i stedet for at man skal vælge en kunde og så trykke på en knap, for at få vist informationerne. Det er også blevet specificeret, når der vælges et nyt element i boksen, så skal der køres en bestemt metode. Det er angivet i boksens "*OnSelectedIndexChanged*" property. Det er lavet så listen altid vil indeholde et element med teksten "*[—- Vaelg en kunde —-]*" Dette element kommer til at stå på plads 0 og når dette element er valgt vil der ikke blive vist noget information. Følgende kode viser hvad der sker når der bliver valgt et nyt element i boksen:

```
Protected Sub listBoxCustomer_SelectedIndexChanged(sender As
    Object, e As EventArgs)
    Session(indexSessionVarName) = listBoxCustomer.SelectedIndex

    If listBoxCustomer.SelectedIndex <> 0 Then
        setFrontpageData()
        enableBtns()
    Else
        clearFrontpageData()
        disableBtns()
    End If
End Sub
```

Det første der sker, er at det indeks nummer som det valgte element i listen har, bliver gemt i en sessions variabel. Denne variabel vil blive overskrevet, hver gang der bliver valgt et nyt element i listen. Grunden til at indeks nummeret bliver gemt i en session variabel, er for at de andre sider der også indeholder en boks med en liste af CapNordic's kunder, vil

vide at der er valgt en kunde og hvilken en der er valgt. På denne måde, er det ikke nødvendigt for brugeren at huske at vælge den samme kunde, når han eller hun går ind på en af de andre sider og skal lave en licens for den kunde der er valgt. Derefter sker der en af 2 mulige ting, alt efter om det valgte element er element nul eller ej. Hvis det valgte element ikke er nummer 0, så vil metoden "*setFrontpageData()*" blive kørt og de knapper man må trykke på, når der er valgt en kunde vil blive aktiveret. Hvis det valgte element er 0, så vil metoden "*clearFrontpageData()*" blive kørt og de knapper man kun må bruge når der er valgt en kunde, vil blive deaktiveret. I metoden "*setFrontpageData()*" vil der først blive oprettet et objekt af typen "*DbFunctions*", så der kan hentes en datatabel op fra databasen, med information omkring den kunde der er valgt. Det kan ses på følgende kode fra metoden:

```
Dim getData As DbFunctions = New DbFunctions()
Dim costumerDt As DataTable =
    getData.getDataTable(CType(Session(sessionConnVarName),Connection),
    sqlQuery)
```

Det er specificeret i strengen "*sqlQuery*", hvilken information der skal hentes op og for hvilken kunde. Når informationen er hentet op, så bliver det taget ud af datatabellen og lagt ind i de respektive controls, hvor det skal blive vist. Følgende eksempel viser hvordan det bliver håndteret, når noget information bliver lagt ind i en tekst boks:

```
If Not IsDBNull(costumerDt(0).Item("CustomerName")) Then
    txtBoxCompanyName.Text = costumerDt(0).Item("CustomerName")
    Else txtBoxCompanyName.Text = ""
```

Det er ikke altid at en kunde i databasen, har noget information der skal i en tekst boks, så derfor bliver teksten i en tekst boks sat til at være tom, hvis den tekst der skulle stå i tekst boksen har en "*NULL*" værdi fra databasen. Er der derimod noget tekst, så vil det blive vist i tekst boksen. Det næste stykke kode viser hvordan det bliver håndteret, når noget information skal ligges ind i en check-boks control:

```
chkBoxMainLicense.Checked = costumerDt(0).Item("Hovedlicens")
```

De værdier fra databasen der skal ligges ind i en check-boks, er enten "*True*" eller "*False*", så den værdi bruges til at angive om der skal være sat et check-mærke i check-boksen.

Når metoden "*clearFrontpageData()*" bliver kørt, vil al information omkring en kunde blive fjernet så de står tomme. Følgende er et eksempel på hvordan det sker for tekst-bokse, ved at sætte teksten til at være en tom streng:

```
txtBoxCompanyName.Text = ""
```

For check bokse vil det ske ved at, sørge for at der ikke er vist noget check-mærke i check boksen, det sker på følgende måde:

```
chkBoxMainLicense.Checked = False
```

Den højre side indeholder en boks med 5 faner, "*Generelt*", "*Version*", "*Kontaktperson*", "*Konsulenter*" og "*Ny kunde*". Fanerne "*Generelt*", "*Version*", "*Kontaktperson*" og "*Konsulenter*", bliver brugt til at vise forskellige kategorier af information om en valgt kunde. Hver af disse faner kan indeholde labels, tekst-bokse, check-bokse, drop-list bokse og knapper. Labels bliver brugt til at indeholde information, omkring hvilken information en tekst-boks, drop-list boks eller check-boks indeholder. Følgende er et eksempel på hvordan en simpel tekst-boks i de 4 faner er sat op:

```
<asp:TextBox ID="txtBoxCompanyName" runat="server"
    Width="200px"></asp:TextBox>
```

De simple tekst-bokse er tekst-bokse på en med en enkelt linie tekst og som det ses, så har de fået et id så server koden kan tilgå controllen og de har fået sat en specifik bredde på 200 pixels. Der findes også nogle tekst-bokse som kan vise flere linier tekst, der viser følgende kode hvordan de er sat op:

```
<asp:TextBox ID="txtBoxLicenseName" runat="server" Rows="7"
    TextMode="MultiLine" Width="200px"></asp:TextBox><br /><br />
```

Her er bredden også sat til 200 pixels, det er der udover også specificeret, at der skal vises flere linier i tekstboksen, ved at sætte "*TextMode*" til at være multiline. Man kan bestemme hvor mange linier der skal blive vist ad gangen i tekst-boksen, ved at angive det ønskede antal linier i property'en "*Rows*". Skulle der være flere linier end hvad der kan blive vist i tekst-boksen, så vil der komme en vertikal scroll-bar frem, så man kan scrolle op og ned i teksten.

En drop-list boks vil blive lavet med et id, en højde som angiver hvor høj den boks der viser det valgte element skal være, en bredde og så også et antal elementer, som der kan vælges imellem i drop-list boksen. Følgende kode er et eksempel på en drop-list boks:

```
<asp:DropDownList ID="dropListProduct" runat="server" Height="17px"
    Width="208px">
    <asp:ListItem>workflow</asp:ListItem>
    <asp:ListItem>documents</asp:ListItem>
    <asp:ListItem>exchangegateway</asp:ListItem>
</asp:DropDownList>
```

En check-boks bliver sat meget simpelt op som det ses af følgende kode udsnit:

```
<asp:CheckBox ID="chkBoxMainLicense" runat="server"
    Text="Hovedlicens" />
```

Som det ses, så har en check-boks også et id, ligesom alle andre controls, der udover kan man sætte en tekst til et check-boks. Denne tekst er en indbygget label, så brugeren kan se hvad de enkelte check-bokse repræsenterer. Da check-bokse har en inbygget label, så er der ikke sat labels ved siden af dem, men kun til tekst-bokse og drop-list bokse.

Hver af de 4 faner har en opdater knap, hvis der bliver trykket på denne knap så vil den opdatere den information der står i den valgte fane, så har man ændret noget information under fanen "*Generelt*" og trykker på knappen "*Opdater*" i den fane, så vil det blive gemt ned i databasen for den valgte kunde. Følgende udsnit viser hvordan en "*Opdater*" knap kan være sat op:

```
<asp:Button ID="btnUpdateGeneral" runat="server"
    OnClick="btnUpdateGeneral_Click" Text="Opdater" />
```

Det tekst der er givet til knappens "*OnClick*" property, er navnet på den metode der bliver kørt når der bliver trykket på knappen. I dette tilfælde er det metoden "*btnUpdateGeneral_Click*" og i denne metode vil den opdatere information fra fanen "*Generelt*". Følgende kode udsnit viser hvad der sker i metoden "*btnUpdateGeneral_Click*":

```
Dim dbF As DbFunctions = New DbFunctions()
dbF.NonQuery(CType(Session(connSessionVarName),Connection),
    sqlQuery)
```

Der bliver oprettet et objekt af klassen "*DbFunctions.VB*", så det er muligt at bruge de nødvendige metoder, til at få forbindelse til databasen. Da der kun skal gemmes data ned i databasen og der ikke er brug for at få noget sendt tilbage, så bliver metoden "*NonQuery*" brugt. Strengen "*sqlQuery*" indeholder den update query, der bruges til at opdatere en kunde, den har også den information, der skal opdateres og for hvilken kunde det skal gøres. Som sagt er der en opdater knap på hver af de 4 faner, de har hver deres metode der bliver kørt, når en af dem bliver klikket på. Dette er for at sikre, at kun den information på knappens fane, bliver opdateret i databasen og den eneste forskel i de metoder er derfor den information der er i strengen "*sqlQuery*". Udover knappen "*Opdater*", så har fanen "*Generelt*" en ekstra knap. På fanen "*Generelt*", er det en "*Slet kunde*" knap. Når der bliver trykket på denne knap, så bliver den valgte kunde slettet. Følgende kode viser præcist hvad der sker, når knappens "*OnClick*" metode "*btnDeleteCustomer_Click*" bliver kørt:

```
Dim dbF As DbFunctions = New DbFunctions()
dbF.NonQuery(CType(Session(connSessionVarName), Connection), query)
```

```
listBoxCustomer.Items.RemoveAt(listBoxCustomer.SelectedIndex)

listBoxCustomer.SelectedIndex = 0

Session(sessionCustomerVarName) = listBoxCustomer.SelectedIndex

clearFrontpageData()
disableBtns()
```

Her bliver der oprettet et objekt af typen "*DbFunctions.VB*", så metoden "*NonQuery*" kan blive brugt til at ændre noget i databasen. Der bliver sendt en streng "*query*" med, som indeholer den sql forespørgsel der sletter den valgte kunde i databasen. Derefter bliver den valgte kunde fjernet fra den boks der indeholder listen over kunder. Når dette er sket så sættes det valgte indeks til at være 0, så der ikke er valgt nogen kunde. Det nye indeks, bliver gemt i sessions variablen, metoden "*clearFrontpageData()*" bliver kørt så der ikke står noget information om den slettede kunde på forsiden og til sidst, så bliver der deaktiveret nogle knapper, som det ikke må være muligt at trykke på, når der ikke er valgt nogen kunde.

Den sidste af de 5 faner på forsiden, er fanen "*Ny kunde*", i denne fane kan man oprette en ny kunde. Fanen indeholder en tekst boks, hvor man kan indtaste navnet på den nye kunde og så har den en knap "*Opret kunde*", som opretter den nye kunde. Når der trykkes på knappen, så bliver metoden "*btnNewCustomer_Click*" kørt. I metoden bliver der sendt en sql forespørgsel ned til databasen som opretter en ny kunde med det navn brugeren har indtastet i tekst boksen. Følgende kode viser hvad der ellers sker i metoden:

```
txtBoxNewCustomer.Text = ""
listBoxCustomer.Items.Clear()

listBoxCustomer.Items.Add(New ListItem("[---- Vaelg en kunde
    ----]", "[---- Vaelg en kunde ----]"))

Dim dtListBox As DataTable =
    dbF.getDataTable(CType(Session(connSessionVarName),Connection),
    query)

For Each customer As DataRow In dtListBox.Rows
    Dim li As ListItem = New ListItem(customer("CustomerName"),
        customer("ID"))
    listBoxCustomer.Items.Add(li)
Next
```

Den tekst boks hvor kunde navnet var indtastet, bliver sat til at være tom og boksen med listen over alle kunderne bliver ryddet. Så bliver der

oprettet et element, som vil være på plads 0 i listen og altså det element der kan vælges når der ikke skal være valgt en kunde. Derefter bliver der udført en sql forespørgsel i databasen, der henter alle kunderne op i en datatabel. En for-løkke kører så gennem hele datatabellen og får lagt alle kunderne ind i boksen, så den kommer til at indeholde en liste med alle de kunder der er, inklusiv den nye kunde der lige er blevet oprettet. Efter alt det er sket, så vil det valgte indeks blive sat til 0, som vil blive gemt i sessions variablen, alle de controls der kan indeholde information om en kunde, bliver ryddet af metoden "*clearFrontpageData()*" og de knapper der ikke må være brugbare, når der ikke er valgt nogen kunde bliver deaktiveret.

Når en bruger kommer ind på siden eller der sker et post-back, så vil metoden "*Page_Load()*" blive kørt. Denne metode bruges til at bestemme hvilke ting der skal ske, når siden bliver vist. I denne metode bliver der til at starte med, kigget på om brugeren har logget ind og at der er blevet gemt en sessions variabel, som kun bliver gemt hvis der er logget ind. Hvis den variabel ikke findes, så vil brugeren blive sendt til Login siden, følgende kode viser hvorden det foregår:

```
If Session(connSessionVarName) Is Nothing Then
    Response.Redirect("Login.aspx")
```

Er der logget ind, vil den gå videre og metoden vil så kontrollere om det er første gang siden bliver vist, altså om brugeren er gået ind på web-siden, eller om det er et post-back foresaget af et tryk på en knap eller lignende, det bliver kontrolleret på følgende måde:

```
If Page.IsPostBack = False Then
```

Grunden til at denne kontrol bliver lavet, er fordi første gange en bruger går ind på siden, så skal den hente alle kunderne op fra databasen og ligge dem ind i boksen, der skal indeholde listen af kunder. Da det ikke er nødvendigt at gøre hver gang der bliver trykket på en knap, så sker det kun hvis det ikke er et post-back. Hvis der på en af de andre web-sider er blevet valgt en kunde og brugeren så går ind på forsiden, så er det også her, at det valgte indeks bliver sat til den kunde, der var valgt på den forige web-side, gennem den sessions variabel der bliver gemt. Igen er det ikke nødvendigt at sætte det valgte indeks hver gang der sker et post-back.

3.3.5 License.ASPX

Denne side er ligesom forsiden, opdelt i 2 dele, en venstre del med en boks, der indeholder en liste med alle CapNordic's kunder, der er lavet på sammen måde som på forsiden og en højre side der består af en tekst boks, med flere linier hvor licensnavnet for den valgte kunde vil blive vist. følgende kode viser hvordan tekst boksen er sat op:

```
<asp:TextBox ID="txtBoxLicenseName" runat="server" Rows="7"
    TextMode="MultiLine" Width="200px"
    ReadOnly="True"></asp:TextBox>
```

Som det ses så er tekst boksen sat til at have flere linier og til at vise 7 linier ad gangen. Tekst boksens "*ReadOnly*" property er sat til at være "*True*", på denne måde er det sikret, at brugeren kan se teksten der bliver vist, men ikke ændre den. Grunden til dette er blevet gjort, er fordi brugeren ikke ved et uheld må ændre i teksten, for ellers vil den licens fil der bliver lavet, ikke passe sammen med vedligeholdelses filer og modul filer, der bliver lavet til det samme firma. I højre side er der også mulighed for at vælge hvilken version filen skal laves til, det er blevet gjort ved at bruge en "*RadioButtonList*", her er det muligt at angive flere radio-button elementer, et til hvert af Workflow versionerne og det vil kun være muligt at vælge en af dem ad gangen. Denne control er sat op på følgende måde:

```
<asp:RadioButtonList runat="server" ID="rdBtnVersions">
    <asp:ListItem>Workflow V5</asp:ListItem>
    <asp:ListItem>Workflow V8</asp:ListItem>
</asp:RadioButtonList>
```

Det sidste der er i højre side er 2 knapper, den ene "*Create license file*", skal bruges af brugeren til at oprette hovedlicens filen, når der er valgt en kunde, den anden knap er "*Download file*", som skal bruges af brugeren hvis han/hun gerne vil downloade filen lokalt på sin egen computer, efter at licens filen er blevet lavet. Når der bliver trykket på knappen "*Create license file*" så udføres knappens "*OnClick*" event, hvor det er blevet bestemt at metoden "*btnCreateLicense_Click*" skal køres. Når metoden bliver kørt, så bliver licensnavnet fra tekst boksen gemt i en streng, og alt efter hvilken version der er blevet valgt i "*RadioButtonList*", bliver der enten gemt "*V5*" eller "*V8*". Der bliver så oprettet en tekst streng "*filePath*", som skal indeholde lokationen på det sted hvor filen er gemt på serveren og der bliver oprettet et objekt af typen "*CreateFile.VB*". Følgende kode viser hvad der sker til sidst i metoden:

```
If cf.createLicenseFile(licenseName, version, filePath) = True Then
    Session(sessionFilePathVarName) = filePath
    btnDlFile.Visible = True
Else
    Session(sessionFilePathVarName) = Nothing
End If
```

Som det ses, så er der lavet en if-sætning hvor der kan ske 2 udfald, alt efter hvilket resultat metoden "*createLicenseFile*" giver. Licensnavnet, Workflow versionen og strengen "*filePath*", bliver sendt med som parametre

i metoden. Strengen "*filePath*" er en reference, så den vil få en tekst fra metoden "*createLicenseFile*". Hvis metoden returnerer "*True*" så er hovedlicens filen succesfuldt blevet lavet og den lokation der nu står i strengen "*filePath*", vil blive gemt i en sessions variabel og knappen "*Download file*" vil blive gjort synlig for brugeren. Hvis metoden returnerer false, så er filen ikke blevet lavet og knappen vil ikke bliver vist. For en sikkerhedsskyld vil sessions variablen, til at gemme lokationen på filen, blive sat til at være tom. Dette bliver gjort, fordi man således ikke kan komme ind på siden "*Download.ASPX*", hvis sessions variablen ikke har en lokation. Knappen "*Download file*" har fået angivet, at den skal køre metoden "*btnDlFile_Click* når knappens "*OnClick*" event forekommer. Som det ses af følgende kode, så bliver brugeren sendt hen til siden "*Download.ASPX*", når der bliver trykket på knappen:

```
Protected Sub btnDlFile_Click(sender As Object, e As EventArgs)
    Response.Redirect("Download.aspx")
End Sub
```

Ligesom med "*Frontpage.ASPX*", så har denne web-side også metoden "*Page_Load*" og her bliver det kontrolleret om brugeren har logget ind og dermed har rettighed til at komme ind på siden. Er brugeren logget ind, så kigges der på en sessions variabel, om der er valgt en kunde fra en anden side og så vil den kunde også blive valgt, i boksen med listen over kunderne.

3.3.6 Maintenance.ASPX

Når brugeren går ind på siden, så vil sidens "*Page_Load*" blive udført, Her bliver det, ligesom med forsiden og licens siden, kontrolleret om brugeren er logget ind, hvis han/hun er det, så vil boksen med kunderne blive fyldt med alle de kunder der findes i databasen og hvis der er valgt en kunde fra en af de andre sider, så vil den kunde blive sat til at være valgt på denne side. Generelt så er denne side lavet utroligt meget ligesom web-siden "*License.ASPX*", den eneste forskel er at, i stedet for en tekst boks til at vise flere linier, så er der en tekst-boks der er beregnet til at indtaste/vælge en dato. Følgende kode viser hvordan den tekst boks er sat op:

```
<asp:TextBox ID="txtBoxDate" runat="server"
    TextMode="Date"></asp:TextBox>
```

Det eneste der adskiller denne tekst boks fra en almindelig tekst boks, er at "*TextMode*" er sat til at være en dato. Dette gør, at når en bruger klikker i tekst boksen, så vil der komme en kalender frem, der gør det muligt for brugeren at vælge en ønsket dato. Denne dato skal bruges til at angive hvornår et produkt skal fornyes næste gang. Der udover så har siden "*Maintenancen.ASPX*" en knap "*Opret maintenance fil*", i stedet for "*Opret licens*

file", som på siden "*License.ASPX*". Der sker dog stort set det samme i de metoder der bliver udført, når knapperne bliver trykket på. Når der bliver trykket på knappen "*Opret maintenance fil*" så bliver metoden "*btnCreateMaintenance_Click*" kørt. For at få oprettet en vedligeholdelseslicens fil, så skal der også bruges et licensnavn, til den kunde filen skal laves til og da siden "*Maintenance.ASPX*" ikke indeholder et licensnavn på den valgte kunde, så skal det først hentes op fra databasen. Udover det så bliver metoden "*createMaintenanceFile*" fra klassen "*CreateFile.VB*" kørt, som det ses af følgende kode udsnit fra "*btnCreateMaintenance_Click*":

```
If cf.createMaintenanceFile(licName, maintenanceDate, version,
    filePath) = True Then
    Session(sessionFilePathVarName) = filePath
    btnDlFile.Visible = True
Else
    Session(sessionFilePathVarName) = Nothing
End If
```

Her bliver de nødvendige parametre, der skal til for at lave en vedligeholdelseslicens fil sendt med og alt efter om filen bliver lavet succesfuldt, så vil det være muligt for brugeren af se knappen "*Download file*" og kunne gå til siden "*Download.ASPX*", hvis filen ikke bliver lavet, så vil det ikke være muligt for brugeren at gå til "*Download.ASPX*" siden.

3.3.7 Modules.ASPX

Ligesom med "*Frontpage.ASPX*", "*License.ASPX*" og "*Maintenance.ASPX*", så vil web-sidens "*Page_load*" blive kørt og her vil det også blive kontrolleret om brugeren har logget ind. Er brugeren ikke logget ind vil han/hun blive sendt hen til login siden. Er brugeren logget ind, så vil alle kunderne blive hentet op fra databasen og tilføjet den boks, der skal indeholde listen over kunderne. På denne side skal der modsat de andre sider, også hentes noget mere op fra databasen, det skal nemlig være muligt på denne side, at vælge hvilke moduler en kunde skal have adgang til. Derfor skal alle modulerne hentes op fra databasen til en datatabel og de bliver lagt ind i et "*GridView*" Følgende kode viser hvordan det foregår:

```
gvModules.DataSource = dt
gvModules.DataBind()
```

Som det kan ses på koden, så bliver datatabellen "*dt*" sat til at være gridview'et "*gvModules*" data source og gridview'ets "*DataBind*" metode bliver så kørt. På denne måde så blive alle modulerne i datatabellen tilføjet som elementer i gridview'et.

Denne web-side består af 2 hoveddele, en venstre side, som indeholder en boks, der skal indeholde en liste af kunderne hos CapNordic A/S, denne

er sat op på samme måde som på de forrige sider. Den højre side indeholder en "*TabContainer*" med 2 faner, "*Create file*" og "*Add module*". I fanen "*Create file*" er der 4 tekst bokse, de skal bruges til at angive en værdi, for hvor mange brugere, dokumenter, indbakker og statistikker en kunde skal have eller skal have udover, hvad de allerede har. Følgende kode fra "*Module.ASPX*" er et eksempel på hvordan de 4 tekst bokse er sat op:

```
<asp:TextBox ID="txtBoxLicensUsers" runat="server"
    TextMode="Number" Width="100px"></asp:TextBox>
```

De har præcis som alle andre controls fået et "*ID*", som kan bruges til at få fat i controllen og dens information, fra noget server kode. De 4 tekst bokse har fået angivet en specifik bredde på 100 pixels, så de har samme størrelse og de har fået sat "*TextMode*" til at være "*Number*", altså tal. Grunden til at de 4 tekst bokse er sat til at indeholde tal, er fordi de, som sagt, skal bruges til at indtaste tal-værdier, til hvor mange brugere, dokumenter, inbokse og statistikker en kunde skal kunne have.

Fanen indeholder også en check boks, som skal bruges til at angive om de værdier der er indtastet i de 4 tekst bokse, skal ligges til det antal af brugere, dokumenter, indbakker og statistikker en kunde allerede har eller om det skal være det totale antal. Følgende kode viser hvordan denne check boks er sat op:

```
<asp:CheckBox ID="chkBoxAddNumbers" runat="server" Text="Add
    LicensUser, LicensDocuments, etc. to current values. Requires
    wf-client &gt;= 843" />
```

Denne check boks er meget simpelt sat op, den har et "*ID*" og så har den fået angivet en tekst, denne tekst er en forklarende tekst der kommer til at stå på højre side af selve check boksen og den vil give brugeren en idé om hvad denne check boks kan bruges til.

Som tidligere nævnt, så bliver alle modulerne lagt ind i et gridview og dette gridview ligger i fanen "*Create file*", følgende kode viser hvordan dette gridview er sat op:

```
<asp:GridView ID="gvModules" runat="server"
    AutoGenerateColumns="False" DataKeyNames="module_db_name">
    <Columns>
        <asp:BoundField DataField="module_name" HeaderText="Module"
            ReadOnly="True" />
        <asp:BoundField DataField="module_db_name"
            HeaderText="Module db name" ReadOnly="True"
            Visible="False" />
        <asp:TemplateField HeaderText="Activate">
            <ItemTemplate><asp:CheckBox ID="ActivateModule"
                runat="server" /></ItemTemplate>
        </asp:TemplateField>
```

```
    </Columns>
</asp:GridView>
```

Det er blevet specificeret at "*AutoGenerateColumns*" skal være "*False*", på denne måde så opretter den ikke nogle default kolonner, men har kun de kolonner der angivet den skal oprette. Der udover har den fået sat at alle de celler der er i kolonnen "*module_db_name*", fra den datatabel der bliver sat til at være data source, skal være "*DateKeyNames*". Grunden til at denne kolonne er sat som data keys, er fordi den kolonne i gridviewet, ikke skal være vist for brugeren og når den ikke bliver vist, så kan man ikke få fat i den tekst der bliver sat ind i cellerne, hvis de ikke er angivet som data keys. Selve gridviewet består af 3 kolonner. Den første kolonne, er en kolonne af typen "*BoundField*", denne type gør det muligt at få bundet data fra gridview'ets data source, til denne kolonne. Ved at sætte kolonnens "*DataField*" property, til at være "*module_name*", så ved gridview'et at værdierne der er i cellerner fra kolonnen "*module_name*", skal ligges over i denne kolonne. Den metode der er angivet svarer til en kolonne, der bliver lagt ind i den datatabel, som gridviewet får som data source. Ved at sætte kolonnens "*HeaderText*" property så kan man give et navn til kolonnen, som vil blive vist øverst på kolonnen. Til sidst er det angivet at "*ReadOnly*" skal være "*True*", på denne måde er det sikret at brugeren ikke kan ændre nogen af cellernes værdier i kolonnen. Den anden kolonne er også af typen "*BoundField*", da denne kolonne også får værdier fra en kolonne, fra data sourcens datatabel. Denne kolonne er stort set sat op på samme måde som den forrige kolonne, den har fået angivet en anden kolonne fra datatabellen i dens "*DataField*" property, et andet navn i "*HeaderText*" og så har den fået sat property'en "*Visible*" til at være "*False*", når det er sat, så vil den ikke blive vist til brugeren. Grunden til at denne er sat til ikke at blive vist, er fordi brugeren ikke har brug for at få vist hvad der står i den kolonne, men det skal være muligt at få fat i de værdier og for at der ikke skal laves et database kald og få hentet den værdi op fra databasen, for hver modul der bliver valgt, så er det nemmere at hente det ud af denne kolonne. Den sidste kolonne er af typen "*TemplateField*", denne type kolonne gør det muligt at indsætte andet i en kolonne end tekst og tal, i dette tilfælde er der blevet indsat en check boks. Så for hver række der bliver sat ind i de 2 andre kolonner fra data sourcen, så vil der blive oprettet en celle med en check boks. Grunden til at denne kolonne er lavet, er for at have en måde, så brugeren kan vælge om et modul skal aktiveres hos kunden eller ej.

Da listen af moduler kan fylde en del, så er gridview'et blevet lavet i et panel. Følgende kode viser hvordan dette panel er blevet sat op:

```
<asp:Panel ID="gvModulesPnl" runat="server" Height="175px"
    ScrollBars="Vertical">
```

Dette panel har fået sat en fast højde på 175 pixels og har fået angivet at den skal have en vertikal scroll-bar. På denne måde vil selve web-siden ikke blive alt for lang, hvis listen af modulerne er meget lang og brugeren vil have mulighed for at scrolle op og ned i panelet så han/hun har mulighed for at vælge alle moduler.

I fanen "*Create file*", er der også en "*RadioButtonList*", hvor det er muligt at vælge om modul licens filen skal laves til Workflow version 5 eller version 8, ligesom på web-siderne "*License.ASPX*" og "*Maintenance.ASPX*". Der udover er der også 2 knapper, en "*Download*" knap som er sat op på samme måde som på "*License.ASPX*" og "*Maintenance.ASPX*". Den anden knap, "*Create module file*", vil køre metoden "*btnCreateModuleFile_Click*" når brugeren klikker på den. Her vil der blive hente information ud af de controls der er i fanen, som skal bruges til at oprette modul licens filen. Følgende kode fra metoden "*btnCreateModuleFile_Click*" viser et eksempel på hvordan det foregår for de 4 tekst bokse:

```
Dim licensUsers As Integer
If txtBoxLicensUsers.Text.Equals("") Then
    licensUsers = 0
Else
    licensUsers = CType(txtBoxLicensUsers.Text, Integer)
End If
```

Der bliver oprettet en variabel af typen "*Integer*", som skal indeholde den værdi der er blevet indtastet, hvis der er indtastet noget. Hvis brugeren ikke har indtastet noget, så vil tekst boksen være tom og så vil variablen blive sat til at være 0, på denne måde vil der ikke blive ændret noget ude hos kunden. Er der derimod indtastet noget, så vil værdien fra tekst boksen, blive lavet om til en integer og lagt ind i variablen. Indtaster kunden selv 0 i tekst boksen, så vil der stadig ikke blive ændret noget ude hos kunderne, da det kun er værdier større end nul hvor der vil ske en ændring hos kunden. Følgende kode viser hvordan værdien fra check boksen bliver taget:

```
Dim addNumbers As Boolean = chkBoxAddNumbers.Checked
```

Her bliver variablen sat til den værdi check boksen har, så er der markeret i check boksen, så vil variablen få værdien "*True*". Er der derimod ikke markeret i check boksen, så vil variablen få værdien "*False*". For at sikre at den korrekte encoding bliver brugt til at lave licens filen, så bliver der lavet en streng, som skal indeholde hvilken version den skal laves til, det kan ses på følgende kode hvordan det gøre:

```
Dim version As String = ""

If rdBtnVersions.SelectedValue.Equals("Workflow V5") Then
    version = "V5"
```

```
ElseIf rdBtnVersions.SelectedValue.Equals("Workflow V8") Then
    version = "V8"
End If
```

For at vide hvilke moduler der skal gives adgang til hos kunden, så bliver der oprettet en "*ArrayList*", som skal indeholde værdier, der angiver hvilke moduler skal åbnes op for hos kunden. Dernæst går metoden ind i en forløkke, som går igennem hele gridviewet for at finde ud af hvilke moduler brugeren har markeret, til at skulle aktiveres hos kunden. Følgende kode viser hvordan det foregår:

```
Dim moduleList As ArrayList = New ArrayList

For Each row As GridViewRow In gvModules.Rows
    Dim activated As CheckBox = row.FindControl("ActivateModule")

    If activated.Checked = True Then
        moduleList.Add(gvModules.DataKeys(row.RowIndex).Values(0))
    End If
Next
```

For hver række i gridviewet, vil der blive kigget på, om den check boks der er i den sidste kolonne, er blevet markeret eller ej. Er den blevet markeret, så vil den tilsvarende værdi, fra den usynlige kolonne, som ligger i gridview'ets data keys blive gemt i array listen.

Til sidst vil den valgte kundes licensnavn blive hente op fra databasen og alt det information der er blevet indtastet og valgt i fanen, vil blive sendt ned til metoden "*createModuleFile*" i klassen "*CreateFile.VB*" og hvis det lykkes at lave filen, så vil lokationen på filen blive gemt i en session variabel og knappen "*Download*" vil blive vist for brugeren.

Den anden fane "*Add module*", indeholer 2 tekst bokse og en knap. Den ene tekst boks skal bruges til at indtaste et navn til modulet, som vil blive vist for brugeren, i modul listen under fanen "*Create file*", hvor der kan vælges moduler. Den anden tekst boks skal bruges til at indtaste, hvad modulet hedder i databasen. Grunden til at der er lavet disse 2 tekstbokse, er fordi navnet i databasen, ikke nødvendigvis ser så pænt ud at få vist eller er så beskrivende, så derfor kan brugeren indtaste et mere sigende navn, som kun vil blive brugt til at blive vist for brugere af licensmanageren. Følgende kode er et eksempel på hvordan de 2 tekst bokse er sat op:

```
<asp:TextBox ID="txtBoxModuleName" runat="server"></asp:TextBox>
```

Som det kan ses ud fra kode udsnittet, så er tekst boksen sat så simpelt op som overhovedet muligt.

Knappen er lige som tekst boksene også sat meget simpelt op, som følgende kode viser:

```
<asp:Button ID="btnAddModule" runat="server"
    OnClick="btnAddModule_Click" Text="Tilfoej modul" />
```

Som det kan ses, så er knappens property "*OnClick*" sat til at have navnet på metoden "*btnAddModule_Click*", så når en bruger trykker på knappen, så vil den metode blive kørt. I denne metode, vil teksten fra de 2 tekst bokse blive lagt ind i 2 forskellige variable, som det ses af følgende kode udsnit fra metoden:

```
Dim moduleName As String = txtBoxModuleName.Text
Dim moduleDbName As String = txtBoxWfModuleDbName.Text
```

Strengene i de 2 variable vil blive lagt ind i en streng "*sqlQuery*", som indeholder den sql insert forespørgsel, der gemmer modulet i databasen. Sql forespørgslen i strengen "*sqlQuery*", vil blive sendt ned til metoden "*NonQuery*" i klassen "*DbFunctions.VB*". Til sidst vil alle modulerne fra databasen blive hentet op i en datatabel, som vil blive sat til at være data source i gridviewet under fanen "*Create file*". På denne måde er det sikret at listen indeholder alle modulerne, inklusiv det nye modul der lige er blevet oprettet i databasen. De 2 tekst bokse i fanen "*Add module*" vil desuden blive sat til at indeholde en tom streng, så der ikke står noget i dem mere.

3.3.8 Download.ASPX

Når en bruger har trykket på knappen "*Download file*", på enten "*License.ASPX*", "*Maintenance.ASPX*" eller "*Module.ASPX*" og er blevet sendt hen til denne web-side, så vil sidens "*Page_Load*" metode blive kørt. Følgende kode viser hvad der sker i metoden:

```
Sub Page_Load()
    If Session(sessionConnVarName) Is Nothing Then
        Response.Redirect("Login.aspx")
    ElseIf Session(sessionLocVarName) Is Nothing Then
        Response.Redirect("Frontpage.aspx")
    End If
End Sub
```

Det bliver kontrolleret om brugeren er logget ind, er brugeren ikke det, så vil han/hun blive sendt tilbage til login siden, for at logge ind. Er brugeren logget ind, men har han/hun ikke fået lavet en af de 3 licens filer og trykket på "*Download file*" knappen, så vil brugeren blive smidt tilbage til forsiden. Dette er lavet, så man ikke bare kan gå ind på web-siden ved at indtaste adressen i adresse baren, uden at have lavet en licens fil først.

Det eneste der er på denne side er en knap med teksten "*Tryk her for at downloade filen*", så når folk trykker på denne knap, så vil de downloade

den licens fil de lige har fået lavet på, enten "*License.ASPX*", "*Maintenance.ASPX*" eller "*Module.ASPX*". Når der bliver trykket på knappen, så bliver dens "*OnClick*" event affyret og så bliver metoden "*btnDownload_Click*" kørt. Følgende kode viser hvad der sker i metoden:

```
Dim path As String = Session(sessionLocVarName)

If path.Contains("maintenance.lic") Then
    Response.ContentType = "lic"
    Response.AddHeader("Content-Disposition", "attachment;
        filename=maintenance.lic")
ElseIf path.Contains("module.mod") Then
    Response.ContentType = "mod"
    Response.AddHeader("Content-Disposition", "attachment;
        filename=module.mod")
ElseIf path.Contains("licenses.id") Then
    Response.ContentType = "id"
    Response.AddHeader("Content-Disposition", "attachment;
        filename=licenses.id")
End If

Response.TransmitFile(path)
Response.End()
```

Aller først bliver lokationen på filen taget ud af sessions variablen og så styrer en if-sætning hvilken af 3 mulige scenarier der skal ske, alt efter hvilken licens fil der er blevet lavet. Det er egentligt det samme der sker i de 3 scenarier, bortset fra at det er forskellige filer der bliver sent. Først og fremmest så bliver det specificeret hvilken type fil, det er der skal sendes til brugerens pc. Der er det enten en "*.lic*" fil type hvis det er en vedligeholdelseslicens fil, en fil af typen "*.mod*", hvis det er en modul licens fil eller en fil med typen "*.id*" hvis det er en hovedlicens fil. Derefter bliver der tilføjet en header til det svar der bliver sent til klienten, denne header får angivet 2 strenge, et navn "*Content-Disposition*" og en værdi, som angiver at der kommer en fil med og hvad den fil skal hedde. Der er fil navnet, ligesom fil typen, forskellig alt efter hvilken af de 3 licenser der er blevet lavet. Til sidst bliver filen så sendt til klienten, som et svar, hvilket vil sige at den bliver downloadet ned lokalt på brugerens computer og så stoppes svaret til klienten.

3.3.9 Ajax Control toolkit

Ajax cotrol toolkit er udviklet af Microsoft til ASP.NET og giver nogle udvidelser, til nogle web-controls og udvidet funktionalitet. Ajax control toolkit, har givet mulighed for at tilføje en udvidelse til en tekts boks, så det er muligt at trykke på tekst boksen og at der så vil blive vist en kalender, her

kan brugeren så vælge år, måned og dag helt frit. Denne control udvidelse bliver brugt på web-siden "*Maintenance.ASPX*", der hvor brugeren skal vælge hvilken ny fornyelses dato, der skal sættes for en kundes produkt. Følgende kode fra "*Maintenance.ASPX*" viser hvordan denne udvidelse er blevet tilføjet til tekst boksen:

```
<asp:CalendarExtender ID="txtBoxDate_CalendarExtender"
    runat="server" TargetControlID="txtBoxDate">
</asp:CalendarExtender>
```

I denne control udvidelse er det blevet specificeret, i dens property "*TargetControlID*" hvilken control udvidelsen skal høre til. Det gøres ved at angive ID'et eller navnet på den control, som skal bruge udvidelsen.

Udover udvidelser til de controls der allerede findes til ASP.NET, så giver Ajax control toolkit også mulighed for at bruge nogle controls, som ikke findes normalt i Microsoft visual studio, men som findes i Ajax control toolkit. En af disse, som også er blevet brugt i udviklingen af den web-baserede licensmanager, er "*TabContainer*", den bliver brugt på web-siderne "*Frontpage.ASPX*" og "*Module.ASPX*". Denne control gør det muligt at lave flere faner, som kan indeholde controls. På denne måde, kan man på en visuel måde, inddele en web-side så det ikke er alle controls der bliver vist på en gang. På denne måde kan man få en mere overskuelig side og brugeren har mulighed for at vælge hvilke controls, han/hun har brug for at se alt efter hvad brugeren skal bruge eller lave.

3.3.10 Fejlhåndtering i web-delen

Fejlhåndtering er koncentreret omkring indtastning af værdier i controls, fra brugerens side. Så for at være sikker på at brugeren indtaster de korrekte ting og at der ikke bliver sendt ugyldige værdier ned til generering af licens filerne. Der er blevet brugt en blanding af 2 metoder, for at sikre de værdier brugeren indtaster er af den korrekte type. Den ene, som kun bliver brugt på nogle tekst bokse i "*Module.ASPX*", er at deres "*TextMode*" property er blevet sat til at være "*Number*". Når dette er blevet sat, så vil tekst boksen lave en automatisk validering af det indtastede, for at kontrollere om det er et tal der er blevet indtastet, valideringen forekommer når der sker et "*Post Back*", som når der bliver trykket på en knap. Er værdien i en tekst boks med "*TextMode*" lig med "*Number*", ikke et tal, så vil det "*Post Back*" der er sket blive stoppet og brugeren vil få at vide at der skal indtastes et tal i tekst boksen.

Den anden måde der bliver brugt, er den metode der bliver brugt mest i web-delen. Dette er gjort ved at bruge Validatorer, som skal validere de controls, der skal bruges til at indtaste eller vælge information, der skal bruges i genereringen af licens filerne. Til dette er der gjort brug af 2 forskellige validatorer. Den ene er en "*CustomValidator*" med denne er det muligt at

styre hvad der skal ske i valideringen, så man kan sikre sig at det kun er nogle bestemte værdier der er accepteret. Følgende kode er et eksempel på hvordan en "*CustomValidator*" er blevet sat op, eksemplet er fra web-siden "*Module.ASPX*":

```
<asp:CustomValidator ID="validatorLicInbox" runat="server"
    EnableClientScript="False"
ErrorMessage="Licens inboxes skal vaere et positivt tal."
OnServerValidate="validatorLicInbox_ServerValidate"
ControlToValidate="txtBoxLicensInboxes">*</asp:CustomValidator>
```

Property'en "*EnableClientScript*" er blevet sat til at være "*False*", for at valideringen sker på server-siden, der udover er der så angivet, når valideringen skal ske, så kører den metoden "*validatorLicInbox_ServerValidate*". For at styre hvilken control den enkelte validering hører til, så angives det i property'en "*ControlToValidate*", hvilken control der skal foretages en validering på. Hvis valideringen ikke bliver godtaget, så vil beskeden i "*ErrorMessage*" blive vist i en "*ValidationSummary*" og der hvor man har placeret valideringen, vil der blive vist det tekst, der er angivet mellem start og end tag-elementerne, i dette tilfælde er det en "*". Følgende kode viser hvordan man kan lave en validering i den metode der er angivet i "*OnServerValidate*":

```
Protected Sub validatorLicInbox_ServerValidate(source As Object,
    args As ServerValidateEventArgs)
If IsNumeric(txtBoxLicensInboxes.Text) Then
    If CType(txtBoxLicensInboxes.Text, Integer) > 0 or
        tabModules.ActiveTabIndex <> 0 Then
        args.IsValid = True
    Else
        args.IsValid = False
        hasErrors = True
    End If
Else
    args.IsValid = False
    hasErrors = True
End If
End Sub
```

Det første der bliver kigget på, er om værdien fra tekst boksen er en tal-værdi, dette er lavet som en ekstra sikring, i tilfælde af at tekst boksen ikke selv skulle lave sin egen validering. Hvis det ikke er en tal-værdi, så bliver "*args.IsValid*" sat til at være "*False*", så validerings controllen ved at værdien ikke må godtages og så vil fejl beskeden blive vist i en "*ValidationSummary*". Der udover så går den også kun ind og laver en validering hvis det er en bestemt fane der er valgt. Hvis den indtastede værdi er et tal, så vil det blive kontrolleret om værdien er positiv, da der ikke må indtastes en negativ værdi. Er den postitiv, så vil "*args.IsValid*"

blive sat til at være "*True*", så det vides at værdien er godkendt. Er værdien negativ så vil valideringen ikke blive godkendt.

Den anden type af "*ValidationControl*", er af typen "*RequiredFieldValidator*", følgende kode er et eksempel på hvordan sådan en kan blive sat op:

```
<asp:RequiredFieldValidator ID="validatorVersion" runat="server"
    ControlToValidate="rdBtnVersions" EnableClientScript="False"
    ErrorMessage="Der skal vaelges en
    version">*</asp:RequiredFieldValidator>
```

Denne type af validering bliver brugt til at validere controls, hvor der skal vælges en værdi, så som en liste af "*RadioButtons*". Opsætningen er meget lig, den måde man opsætter en "*CustomValidator*" på. Det er angivet hvilken control der skal valideres i property'en "*ControlToValidate*". Den eneste forskel der egentligt er, er at man i denne validator control ikke skal angive en metode der skal køres når der skal ske en validering, da denne automatisk kigger på om der er valgt et element i listen. Er der ikke det, så vil validering være "*False*" ellers så vil validering være "*True*".

4 Test

I dette kapitel vil jeg komme ind på hvordan jeg har testet de forskellige dele af den endelige web-baserede licensmanager efter den var færdig udviklet.

4.1 Test af Visual Basic klasserne

4.1.1 DbFunctions.VB test

Som tidligere nævnt, så indeholder denne klasse 2 metoder, "*getDataTable*" og "*NonQuery*". Da metoden "*getDataTable*" skal kunne hente noget data op fra databasen, ligge det ind i en datatabel og returnere datatabellen, er der derfor lavet tests, hvor metoden har modtaget en forespørgsel, som skulle give noget data, for at se om den ville returnere de korrekte ting, ud fra hvad der var specificeret i forespørgslen. For at være sikker på at metoden fik hentet det rigtige op, blev det data der var i den returnerede datatabel, sammenlignet med det data der blev vist ved at lave den samme forespørgsel i selve databasen manuelt. Hvis der var forskel i disse 2 ting, så ville der være noget galt med metoden. Testen er blevet lavet ved at sende forespørgsler, der enten skulle hente meget eller lidt data op. I alle de tests der er lavet, har der ikke været forskel på det metoden "*getDataTable*" har returneret og hvad der er blevet vist, ved at lave en manuel forespørgsel direkte i databasen.

Metoden "*NonQuery*" bruges, som der er skrevet tidligere i rapporten, til at ændre information i databasen og når der ikke skal hentes noget op fra databasen. Denne metode er blevet testet meget lig metoden "*getDataTable*", bortset fra at denne metode ikke returnerer noget, så der kan ikke sammenlignes på samme måde. Denne metode har jeg derfor testet, ved at have 2 databaser der var fuldstændig ens og så ville jeg i den ene database ændre information, ved at bruge denne metode og den anden database ville jeg ændre information, ved at skrive forespørgslen manuelt i databasen og hvis de samme ting var ændret i begge databaser, så ville metoden virke korrekt. Hvis de ikke var ens, kunne det betyde at metoden lavede noget forkert og måske sendte forespørgslen forkert ned til databasen. Der har dog ikke været nogle tilfælde, hvor der har været en forskel. De forespørgsler der er blevet lavet har været forskellige i den forstand, at de enten skulle slette, ændre eller oprette noget i databasen og det har enten været en enkelt celle eller flere celler ad gangen.

For at være helt sikker på at de 2 metoder virkede korrekt, har jeg selv siddet og fundet ud af, hvad der skulle vises eller ændres i databasen alt efter hvilken metode der blev brugt og har så set om det jeg selv forventede der ville ske skete. De eneste tidspunkter hvor det jeg forventede der ville ske ikke skete, var fordi der var skrevet en fejl i forspørgslerne, så metoderne virker korrekt og udfører den sql forespørgsel der bliver modtaget.

4.1.2 Connection.VB test

Her er der blevet testet om metoden "*testLogin*" korrekt kunne logge ind i den angivne database, når man gav det korrekte information og at den ikke loggede ind hvis der ikke blev angivet et korrekt brugernavn og kodeord. Testen af denne er lavet, ved selvfølgelig at give metoden den korrekte kombination af kodeord og brugernavn og så se om den kunne få forbindelse til databasen og at den ville gemme det objekt der skulle kunne oprette forbindelse til databasen korrekt. Der udover er det også testet, at hvis den forkerte kombination blev modtaget, så ville den ikke gemme noget objekt der kan oprette forbindelse til databasen. I begge scenarier har metoden opført sig som den skulle, når den henholdsvis skulle logge ind med det korrekte information og hvis den modtog det forkerte.

Der udover er det testet at klassens property "*objCon*", korrekt kunne returnere det objekt der kan oprette forbindelse til databasen. Der er testen lavet ved at sende objektet med ned til metoderne i klassen "*DbFunctions.VB*" og så se om den oprettede forbindelse til databasen som den skulle, så der kunne sendes en sql forespørgsel ned til databasen. Der har ikke været oplevet nogen fejl med oprettelsen til databasen, ud fra det objekt der bliver returneret fra property'en.

4.1.3 CreatFile.VB

Her i denne klasse har jeg testet om metoderne virkede korrekt og gjorde det de skulle. Først og fremmest har det været at teste, om der kunne laves en fil og at den kunne gemmes hvor det var bestemt den skulle gemmes. Det har der ikke været noget problemer med overhovedet for nogen af de 3 filer. Det næste der blev testet, var om metoderne "*createLicenseFile*", "*createMaintenanceFile*" og "*createModuleFile*" kunne ligge det korrekte information ned i filerne, ud fra det de modtog. Der er blevet lavet en del test, hvor de har modtaget vidt forskellig information, så som forskellige licensnavne, datoer, forskellige moduler, antal brugere, dokumenter, indbakker, statistikker osv. Af de 3 licensfiler, er der blevet lavet mange tests, for at se om de kun gemte det korrekte og ikke kom til at ligge noget data derned, der ikke skulle med og det har der ikke været nogen tilfælde af. Det næste der blev testet var om de blev krypteret korrekt. I denne test er der igen for hver af de 3 licens filer, blevet lavet tests med forskellig information. For hver test af en fil er der blevet sammenlignet med en licens fil lavet af den gamle licensmanager og med det samme information, for at se om de var helt ens. Igen har der ikke været nogen tilfælde, hvor der har været nogen forskel. Den sidste type test der har været lavet og nok den vigtigste test af licensfilerne, er om Workflow version 5 og Workflow version 8 har kunnet indlæse filerne korrekt og ikke ville fejle, med noget af det information der stod i filerne. Igen er der her lavet flere tests, ved at have forskellig information i filerne.

Grunden til dette er for at være sikker på, at de læser filerne korrekt og at der ikke er nogle tilfælde hvor de fejler. I alle tilfælde har versionerne korrekt kunne læse en hovedlicens fil og korrekt vidst hvilket firma produktet nu tilhørte, der udover så har de korrekt kunne indlæse en vedligeholdelseslicens fil og ændre datoen for hvornår produktet skal fornyes. Modul filen har også været indlæst korrekt så de moduler der stod til at skulle aktiveres er blevet aktiveret og det antal dokumenter, brugere, indbakker og statistikker der skulle være eller tilføjes, også blev sat korrekt. Det er også blevet testet at man ikke kunne bruge filer der var blevet lavet til forskellige kunder.

Det skal nævnes, at testen af filerne ikke er blevet lavet ude hos de rigtige kunder, men er blevet lavet på en computer hvor det er muligt at teste indlæsning af filerne.

4.2 Test af web-delen

I selve web-delen, har jeg testet om alle controls virker som de skal, og om det server kode der hører til de enkelte web-sider gør hvad de skal. Disse tests er foretaget i browseren "*Google Chrome*".

4.2.1 Login.ASPX test

På denne side har jeg testet, hvad der sker når brugeren indtaster et korrekt login i de 2 tekst bokse der er og hvad der sker hvis brugeren indtaster noget forkert. Da der ikke er nogen begrænsninger for hvad der kan indtastes i de 2 tekst bokse, så har der ikke været grund til at teste om begrænsningerne virker. Hoveddelen af testen på denne side, har været omkring knappen, her skulle det testes, at den metode der står til at blive kørt, når der bliver trykket på knappen bliver kørt og om den tager teksten korrekt fra de 2 tekst bokse. Det er også blevet testet, om den sessions variabel der bliver gemt hvis man logger ind korrekt, rent faktisk kommer til at indeholde det den skal. Det er gjort ved at hente variablen ned, på både denne side og de andre web-sider, for at sikre sig at den indeholdt det samme og det der blev gemt i den. Der har ikke været nogen situationer, hvor det der blev gemt i variablen, ikke var det der blev hentet ned fra den samme variabel på nogen af web-siderne. Der udover skulle det også testes om brugeren blev sendt til "*Frontpage.ASPX*", når der blev trykket på knappen og de korrekte login oplysninger var indtastet i tekst boksene. Den sidste ting der blev testet på siden, var om man blev sendt hen til "*Frontpage.ASPX*", hvis man via adresse baren prøvede at komme ind på login siden selvom man allerede var logget ind.

4.2.2 Frontpage.ASPX test

For at kunne komme ind på denne side, skal man være logget ind, jeg har derfor testet om det har været muligt at komme ind på denne side, ved at

indtaste adressen i adresse baren, uden at være logget ind, det har dog ikke vist sig at være muligt, da jeg så blev sendt tilbage til login siden. Det eneste tidspunkt jeg har adgang til siden, er hvis jeg loggede ind først.

For at teste list boksen, har jeg kontrolleret at de kunder der er vist, også er de samme som dem der er i databasen, der udover har jeg testet, at når man vælger et nyt element i listen, så bliver metoden "*listBoxCustomer_SelectedIndexChanged*" kørt. Når denne metode bliver kørt, så har jeg testet at der bliver gemt en sessions variabel, der indeholder information om hvilken kunde der er valgt, så den også bliver valgt, når man går ind på en af de andre web-sider med en list boks med kunder. Ved at skifte mellem siderne, kan jeg se at det er det korrekte der bliver gemt i sessions variablen, da det er de samme kunder der bliver valgt på de andre sider, som den man vælger på denne side. Der udover har jeg kontrolleret at det information der bliver hentet op fra databasen, når der er valgt en kunde er det korrekte information. Det er gjort ved at sammenligne det der bliver vist i de forskellige controls i alle fanerne, med det der står i databasen. Det der bliver vist i de controls der er på forsiden, er det samme som det der står i databasen. Jeg har også fået kontrolleret om informationen fra databasen bliver lagt i de korrekte controls og der har ikke været noget tilfælde, hvor der blev vist noget forkert information, i nogen af de controls der er.

I fanerne "*Generelt*", "*Version*", "*Kontaktperson*" og "*Konsulenter*", har jeg testet hvad der sker, hvis man ændrer noget i nogle af de controls der er og trykker på en update knap. Ved at sammenligne med det der står i de enkelte controls og hvad der er blevet ændret i databasen, så kan jeg se at, det kun er det information, der står i de enkelte faner, der bliver ændret i databasen, hvis man trykker på en update knap, hvilket også er meningen. Der udover så bliver det der står i en fanes controls, også gemt korrekt ned i databasen og det kan hentes korrekt op fra databasen.

Når der bliver trykket på knappen "*Slet kunde*" under fanen "*Generelt*", så skal den kunde der er valgt, gerne blive slettet, dette er blevet testet ved at oprette nogle falske kunder i databasen, med noget falsk information, som gerne må slettes og så vælge dem i listen af kunder og så trykke slet. Når knappen bliver klikket på, så bliver kunden også slettet fra databasen og fjernet fra listen over kunder, så den virker helt som den skal.

Det sidste jeg har testet på denne side, er under fanen "*Ny kunde*", her skal det være muligt at oprette en kunde, med det navn som man har angivet i tekst boksen og det sker også succesfuldt, når man trykker på knappen "*Opret kunde*". Der bliver oprettet en ny kunde i databasen, med det navn man har angivet i tekst boksen og den nyligt oprettede kunde, vil også blive vist i listen over kunder på siden.

4.2.3 License.ASPX test

Det skal kun være muligt at komme ind på denne side, ved at brugeren har logget ind først, der er derfor blevet testet, ved at indtaste adressen på siden i adresse baren, når man er logget ind og ikke logget ind, for at se hvad der sker. Er der ikke blevet logget ind, så bliver man sendt hen til login siden, for at kunne logge ind. Er der derimod logget ind, så vil man fint komme ind på siden. List boksen i venstre side, er blevet testet på samme måde, som på forsiden, at den har fået alle kunderne op fra databasen, ved at sammenligne selve listen med hvad der er i databasen.

Da den tekst boks der er på denne side, kun skal kunne vise tekst og at det ikke skal være muligt for brugeren at indtaste noget i tekst boksen, så har jeg testet om det var muligt ved at klikke i tekst boksen for at se om man på en eller anden måde ville kunne tilføje tekst, det har dog ikke været muligt. Der udover har jeg testet om det er muligt at vælge flere elementer i listen af radio buttons til workflow versioner, men det har ikke været muligt, så man kan kun vælge 1 version ad gangen.

De sidste ting der er blevet testet er når knappen "*Create license file*" bliver klikket på af brugeren at der så bliver lavet en fil til den kunde der er valgt og at knappen "*Download file*" bliver vist hvis filen bliver lavet. Det aller sidste er så om man vil blive sendt til download siden hvis man klikker på knappen "*Download file*", hvilket man gør.

4.2.4 Maintenance.ASPX test

På denne side er testen foregået på samme måde som på "*License.ASPX*", den eneste forskel der har været i test på denne side er hvordan tekst boksen med datoen er blevet testet. Da denne tekst boks skal indeholde en dato, så har det været vigtigt at få testet om det er muligt at indtaste andet end en dato og hvis det er, at fejlen så bliver fanget så den ikke prøver at oprette en vedligeholdelseslicens fil med noget der ikke er en dato. Testen af dette er blevet foretaget ved at klikke sig ind i tekst boksen og prøve at skrive bogstaver, men da tekst boksen er sat til at have tekst mode "*Date*", så vil den ikke lade en indtaste andet end tal for dage, måneder og år.

4.2.5 Module.ASPX test

Det første der er blevet testet på denne side, er om man kan komme ind på siden hvis man ikke er logget ind, det er lige som med de andre web-sider, blevet gjort ved at indtaste adressen på web-siden i adresse baren, uden at være logget ind og se om man kunne komme ind på siden eller om man blev sendt hen til login siden. Det har ikke vist sig at være muligt at komme ind på siden med mindre man er logget ind.

Det næste der er blevet testet er hvordan de 4 tekst bokse til at indtaste tal for antallet af brugere, dokumenter, indbakker og statistikker opførte sig.

Det skal nemlig ikke være muligt at indtaste andet end tal, man kan sagtens indtaste bogstaver i tekst boksene, men hvis man trykker på en knap eller på anden måde foretager noget så der forekommer et "*Post Back*", så vil tekst boksene selv finde ud af om det er en tal værdi der er indtastet. I de tilfælde hvor det ikke er en tal værdi der er indtastet, så vil de lave en pop-up besked til brugeren der fortæller at der skal indtastes et tal. Er der derimod indtastet et tal, så vil tekst boksene acceptere det der er blevet indtastet. Det skal dog heller ikke være muligt at sende negative værdier videre til fil genereringen, men testen af det vil jeg komme ind på under afsnittet "*Test af fejlhåndtering.*

Den check boks der er på siden under fanen "*Create file*" har det været nødvendigt at teste om de modul filer der blev lavet henholdsvis tilføjede til de nuværende værdier eller satte de totale værdier for brugere, dokumenter, indbakker og statistikker når man markerede i check boksen og når man ikke markerede i check boksen. Ved at se på hvordan værdierne ændrede sig ud fra om der er blevet markeret eller ej i check boksen, så virker det helt som det skal. Er der markeret i check boksen, så bliver værdierne lagt til de værdier der allerede er, men er der ikke markeret i check boksen, så vil værdierne fra filen blive sat som de totale værdier.

Med gridview'et der indeholder listen af moduler, har jeg testet om det er de korrekte modul navne der bliver sendt ned til fil generering og åbnet op for i workflowet alt efter hvilke moduler man angiver der skal aktiveres. Dette er gjort ved at markere for en eller flere moduler og gøre det med skiftende moduler og så holde øje med om de moduler man har markeret i aktiverings kolonnen også er dem der blive åbnet op for. Der har ikke været nogle tilfælde hvor der er blevet åbnet op for andre moduler end dem der er angivet til at skulle aktiveres.

Listen af "*RadioButtons*" er blevet testet på samme måde som på websiderne "*License.ASPX*" og "*Maintenance.ASPX*" og den har virket som den skal ligesom på de 2 andre sider.

De knapper der er i begge faner på denne side, er blevet testet ved at se på om de respektive metoder der skal køres når man klikker på dem bliver kørt og om metoderne henter det korrekte information ned fra de controls de skal og sender det korrekt videre. Det er blandt andet blevet gjort ved at holde øje med om det der blev indtastet i tekst boksene også var det der blev taget ud af dem og hvis det skulle gemmes i databasen om det så var det samme der blev gemt ned. Der har ikke været oplevet nogle tilfælde hvor det der er blevet angivet i de forskellige controls ikke var det metoderne tog og brugte til enten at generere en fil, oprette et nyt modul i databasen og opdatere listen af moduler eller sende en hen til download siden.

4.2.6 Download.ASPX test

På denne side har det været nødvendigt at få testet 2 ting når brugeren prøver at komme ind på siden. Det første er om brugeren bliver sendt hen til login siden hvis han/hun ikke er logget ind og den anden er om brugeren bliver smidt hen til forsiden hvis han/hun ikke har fået lavet en fil fra enten "*License.ASPX*", "*Maintenance.ASPX*" eller "*Module.ASPX*". Det er blevet testet ved at indtaste adressen til denne side i adresse baren. Det er gjort hvor brugeren enten er logget ind eller ikke er logget ind og når brugren er logget ind, så er det gjort når der ikke er blevet lavet en fil og når der er blevet lavet en fil. I de tilfælde hvor brugeren ikke er logget ind, så bliver man sendt korrekt til login siden. Er der blevet logget ind, men ikke blevet lavet en licens fil, så bliver man korrekt sendt hen til forsiden. Det eneste tilfælde er hvor man kan komme ind på siden er når man både er logget ind og lige har lavet en fil, hvilket også er det eneste tidspunkt det skal være muligt.

Den sidste ting der skal testes på denne side er om der bliver downloadet en fil korrekt og om det er den korrekte fil. Dette er testet simpelt ved at lave en fil fra enten "*License.ASPX*", "*Maintenance.ASPX*" eller "*Module.ASPX*" og så gå ind på denne side og hente filen. Passer den fil der er lavet, sammen med den fil der er blevet hentet ned og er det af den korrekte type som den der lige er blevet lavet, så virker det korrekt. De filer der bliver hentet ned gennem browseren passer med den fil der lige er blevet lavet og de indeholder det samme.

4.2.7 Test af fejlhåndtering

Da fejlhåndteringen er lavet af validatorer til web-controls, så har jeg testet om de forskellige validatorer virkede som de skulle og ikke gjorde noget forkert. For eksempel har jeg testet at de validatorer der er til hver af de 4 tekst bokse på web-siden "*Module.ASPX*" til indtastning af antallet af brugere, dokumenter, indbakker og statistikker ville acceptere negative værdier. Det skal nemlig ikke være muligt at sende en negativ værdi ned til generering af filen. Jeg har derfor ved at indtaste negative værdier, set om validatorerne har reageret korrekt og angivet det som en fejl. Der har jeg ikke kunne få sendt nogle negative værdier til generering af modul licens filen. Validatorerne fanger fint fejlen og angiver at det skal være en positiv værdi der skal indtastes i tekst boksene og al kørsel bliver også stoppet så der ikke bliver forsøgt at lave en modul licens fil.

Der er foretaget den samme test af validatoren på web-siden "*Maintenance.ASPX*", her er det blevet testet om validatoren fanger fejlen hvis der står noget der ikke kan konverteres om til en dato og her bliver det også fanget korrekt og angivet at det skal være en dato der skal indtastes.

Jeg har også fået testet at validatorerne der er lavet til list boksene

på siderne "*License.ASPX*", "*Maintenance.ASPX*" og "*Module.ASPX*" angiver at der skal vælges en kunde, hvis der ikke er valgt nogen kunde når den står på det første element i listen som har teksten "*[—- Vaelg en kunde —-]*" og det sker også som det skal.

En lignende test er også lavet for de validatorer der er sat til radio button list på siderne "*License.ASPX*", "*Maintenance.ASPX*" og "*Module.ASPX*" hvis der ikke er valgt nogen version. Der skal nemlig være valgt en version ellers ved fil genererings metoderne ikke hvilken version filerne skal laves til. Jeg har derfor testet hvad der sker hvis man ikke vælger en version og prøver at lave en fil på hver af de 3 web-sider. Validatorne fanger succesfuldt fejlen og angiver at man skal vælge en version før man kan oprette en licens fil.

4.3 Test i andre browsere

Jeg har valgt at teste om licensmanageren virker i de 3 mest brugte browsere: "*Google Chrome*", "*Mozilla Firefox*" og "*Internet Explorer*". Grunden til jeg har gjort det er for at være sikker på at dem der skal bruge hjemmesiden kan bruge den browser de bruger som standard og så der ikke er ting på hjemmesiden som de ikke kan udføre hvis det på en eller anden måde ikke skulle være understøttet. Da hoved testen er foretaget i "*Google Chrome*" og da jeg har testet på samme måde i browserne "*Mozilla Firefox*" og "*Internet Explorer*", så vil jeg i de 2 test afsnit omkring "*Mozilla Firefox*" og "*Internet Explorer*" nøjes med at beskrive hvad der gav nogle andre resultater når jeg testede i dem.

4.3.1 Mozilla Firefox

I denne browser er jeg kun stødt på en ting som ikke virkede korrekt og det var den tekst boks på web-siden "*Maintenance.ASPX*" som skulle vise en kalender når man klikkede i tekst boksen. Den kom af en eller anden grund ikke frem her, men jeg fandt så ud af at den udvidelse som Ajax control kit havde til en tekst boks der er sat til at være med "*TextMode*" lig Calendar, virkede så man også i denne browser ville kunne klikke i tekst boksen og således få en kalender frem så brugeren vil kunne vælge en dato. Der udover har alle andre controls virket som de skulle og alt har reageret efter planen.

4.3.2 Internet Explorer

I denne browser oplevede jeg det samme problem som i Mozilla Firefox, her ville tekst boksen på "*Maintenance.ASPX*" heller ikke vise kalenderen når man klikkede i tekst boksen. Heldigvis virkede udvidelsen fra Ajax control kit også fint i denne browser, hvilket har gjort at man nu i alle 3 browsere vil kunne klikke i tekst boksen på web-siden "*Maintenance.ASPX*".

5 Konklusion

Hele den web-baserede licensmanager er lavet så det passer til de krav CapNordic har stillet. Udover det så har jeg fået positiv respons fra dem når jeg har vist den web-baserede licensmanager frem for dem, så de kunne se hvordan den så ud. Jeg vil derfor sige at projektet har været en succes. Meningen med selve projektet har kun været at få udviklet hjemmesiden og få testet den igennem for at se om det overhovedet kunne lade sig gøre at lave en web-baseret licensmanager. Det næste der så skal ske med denne licensmanager er at den skal op og køre på CapNordic's server, således at de kan komme igang med at bruge den web-baserede licensmanager til at lave de licens filer de har brug for at få lavet. Når den er kommet op og køre, så vil det være en hel del nemmere for alle konsulenterne og cheferne at få lavet de licens filer de har brug for at få lavet.

Efter licensmanageren er kommet op og køre og når der er tid til det, så er der selvfølgelig altid nogle ting der kan videreudvikles på hjemmesiden, såsom noget bedre sikkerhed og lidt pænere design. Det var som tidligere nævnt ikke et krav at det skulle laves på nuværende tidspunkt, men det er klart en af de ting der kan laves i fremtiden.

Jeg føler selv at det har været et sjovt og spændende projekt, i og med at jeg ikke har haft den store viden inden for web-udvikling før, så det har været et projekt der har udviklet mine egne evnet og jeg føler mig nu en hel del mere sikker i hvordan hjemmesider kan laves. Der udover så synes jeg det har været en lærerig proces omkring at jeg har lavet noget og samtidigt hele tiden haft kontakt til firmaet så vi kunne snakke om hvordan tingene skulle laves og undervejs snakke om hvordan det så ud og om de var tilfredse med det.