

1 IT i byggeriet

Projektet tager udgangspunkt i et andelsbølgbyggeri på 40 boliger i Visse, hvor Højgaard & Schultz (HS) og Kuben har indgået en partneringliggende aftale. Denne aftale har medført at brugerne har fået valgmuligheder mht. indretningen af boligen. Det betyder, at brugerne bl.a. kan bestemme hvilken type køkken, gulv, dørhåndtag osv, de ønsker i boligen. Desuden har de haft mulighed for at ændre på placering af vægge og ruminddeling. Brugernes indflydelse på indretningen af boligen har givet anledning til øget pres på især byggelederen, da det er ham, som står for planlægningen og koordineringen af ændringerne. Der ligger en stor koordineringsopgave i, at alle boliger indretningsmæssigt bliver forskellige.

Arbejdet for byggelederen forøges, da køberne ofte kommer med ændringsønsker efter udløb af en fastlagt tidsfrist. Derved opstår der et ressourcespild, som burde undgås. De ændringer, som foretages fra brugernes side, registrerer byggelederen i et mappesystem. Et sådan mappesystem kan være svært at overskue og give anledning til, at en rettelse bliver lavet i det forkerte hus. Desuden kan det give problemer ved evt. bortkomst af mapperne, da de kun findes i èt eksemplar.

Dette leder frem til følgende problemstilling:

Hvordan udarbejdes et system til styring af ændringer, som tager højde for det kommunikationsbehov, der er mellem bruger og byggeleder ?

Til løsning af ovenstående problem udarbejdes først nogen konceptuelle modeller, som beskriver de forskellige dele og processer som skal indgå i det udviklede system.

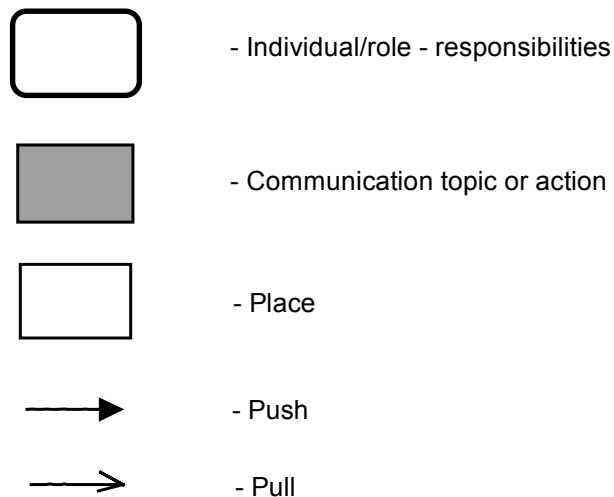
1.1 Konceptuelle modeller

I det følgende beskrives tre forskellige modeller, Workflow, Produktmodel og Sekvensdiagram.

1.1.1 Workflowdiagram

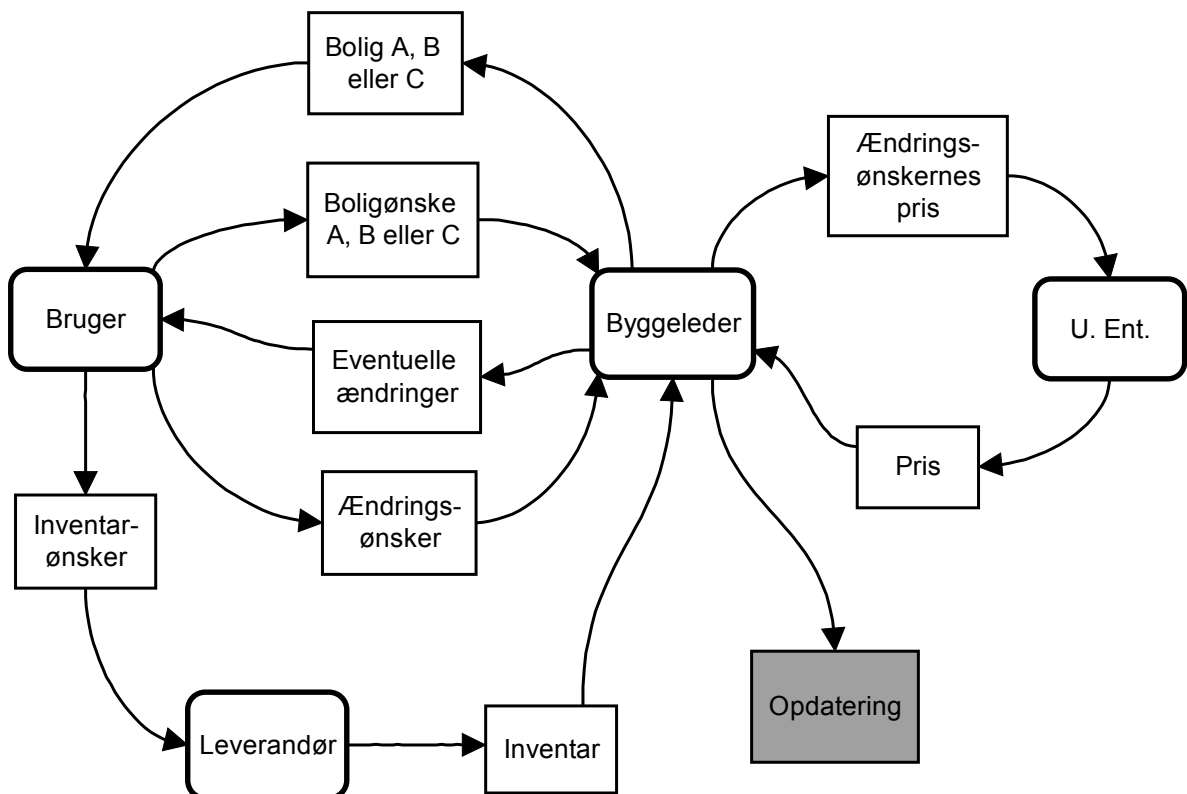
I dette afsnit opstilles 2 workflowdiagrammer for henholdsvis workflowdiagrammet over det aktuelle forløb i projektet samt for løsningsmulighedens forløb i projektet.

Signaturforklaringen til disse diagrammer er[Diversity project]:



1.1.1.1 Workflowdiagram over det aktuelle forløb i projektet

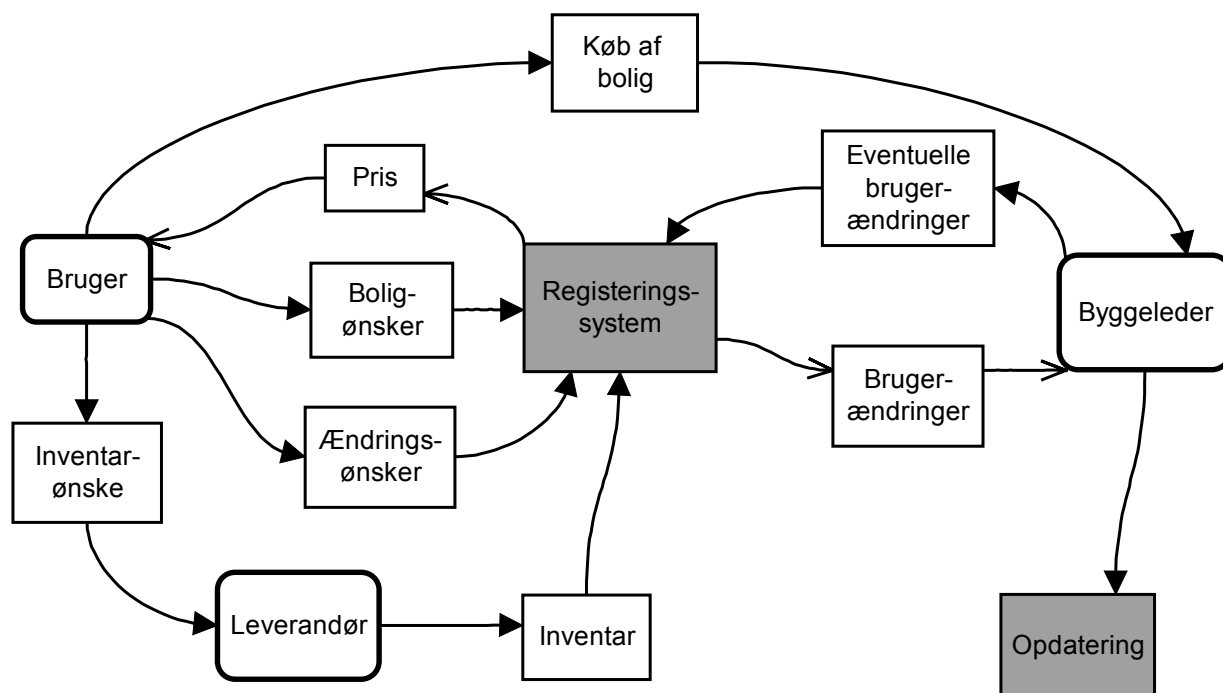
Workflowdiagrammet over det aktuelle forløb i projektet er opstillet i figur 1.



Figur 1. Workflowdiagram over det aktuelle forløb i projektet.

Arbejdsprocessen for det aktuelle projekt er koncentreret omkring byggelederen, hvor størsteparten af brugerens ønsker og ændringer kommunikerer direkte med byggelederen. Da byggelederen både kommunikerer med de forskellige leverandører og de enkelte brugere samt opdateringen af brugernes ønsker, bruges meget af byggelederens tid til disse opgaver.

1.1.1.2 Workflowdiagram over løsningsmulighedens forløb i projektet



Figur 2. Workflowdiagram over det tiltænkte forløb i projektet.

Arbejdsprocessen som viser løsningsmulighedens forløb i projektet, ligger koncentreret omkring registreringsystemet. Kommunikation mellem brugeren og byggelederne forekommer kun ved køb af boliger. Ligeledes forsvinder kommunikationen mellem byggelederen og de enkelte leverandører. Dermed opnår byggelederen mere tid til andre arbejdsområder. I dette projekt belyses brugernes og byggelederens kommunikation omkring registreringsystemet.

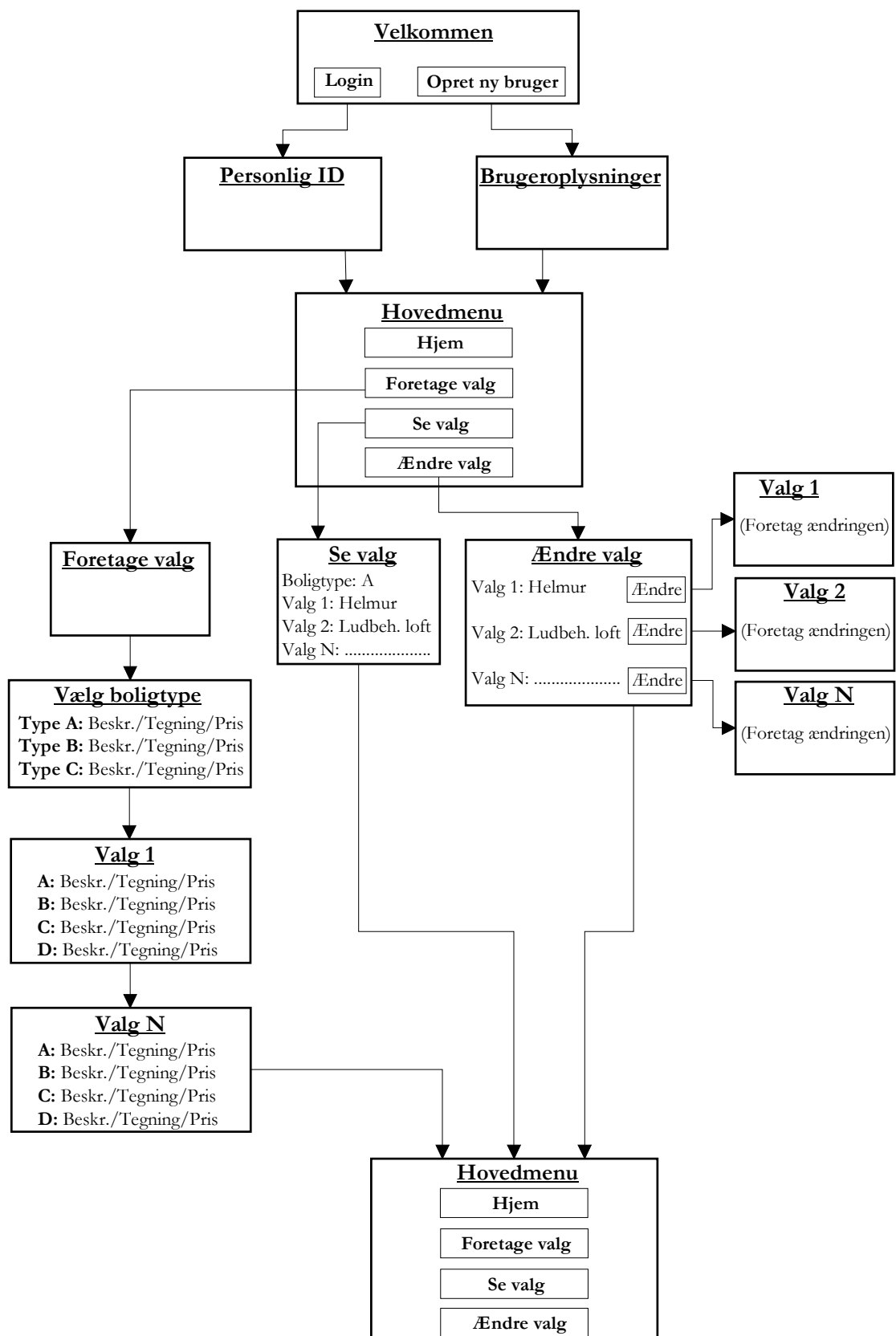
1.1.2 Produktmodel

Produktmodellen opbygges i det følgende. Grundmodellen er som nedenstående figur.

<p><u>Køkken</u> Type. Gulvtype. Vægtype.</p>	<p><u>Lejlighedstype</u> Type: A, B, C Specifikke ændinger Mur v. køkken: hel, halv, ingen Værelse i stuen: ja/nej</p>
<p><u>Badeværelse</u> Type. Gulvtype. Vægtype.</p>	<p>Værelse ved trappe: ja/nej Loft på toilet: hvid beton, træ, ludbeh. træ Resterende loft:: hvid beton, træ, ludbeh. træ Skabe i soveværelse: ja/nej Gulv i bryggers: fliser 1-2-3, Linolium 1-2-3</p>
<p><u>Brugervalg</u> Ovn. Køleskab m. frys. Opvaskemaskine. Vaskemaskine. Tørretumbler.</p>	<p>Resterende gulv: parket 1-2-3, gulvtæppe 1-2-3, Vægge: tapet , malet, vandskuret Døre indvendig: type 1-2-3</p>

Figur 3. Produktmodel.

1.1.3 Sekvensdiagram



Figur 4. Sekvensdiagram

Sekvensdiagrammet, i figur 4, beskriver hvilke sekvenser brugeren udsættes for ved brug af registreringssystemet. Når brugeren logger på hjemmesiden, er der en velkomstsider hvor brugeren enten kan vælge at logge på hjemmesiden eller oprette sig selv i brugerkartoteket. Hvis brugeren i forvejen er oprettet i kartoteket kan vedkommende nøjes med at logge på hjemmesiden vha. brugernavn og password. Hvis det er første gang brugeren besøger hjemmesiden skal vedkommende oprette sig selv ved at indtaste brugeroplysninger, som eksempelvis navn, adresse, telefonnr. mm. Derefter kommer brugeren ind på hovedmenuen, hvor det er muligt at *foretage valg, se valg* eller *ændre valg*. Hvis det vælges at *foretage valg*, guides brugeren igennem alle valgmulighederne. Det første valg som brugeren skal træffe er, hvilken type bolig vedkommende vil have. Dernæst skal brugeren træffe valg som bl.a. omhandler lofttype, gulvtype og vægtype. Fælles for valgene er at de indeholder en beskrivelse, tegning og pris. Brugeren kan løbende se den samlede pris for valgene der er foretaget. Efter at have foretaget valgene vendes tilbage til hovedmenuen. Brugeren kan så se hvilke valg der er foretaget og hvad de koster, ved at vælge *se valg*. Brugeren kan også vælge at *ændre valg*, dette kan være tilfældet hvis prisen på de foretagne valg overstiger det brugeren havde forventet og der for ønsker at vælge om.

1.1.4 Opsamling

Ud fra de to Workflowmodeller ses at byggelederen har mulighed for at bruge tiden mere effektivt, hvis registreringssystemet benyttes.

Produktmodellen beskriver hvilke produkter der skal indgå i systemet. Det bør bemærkes at køkken, badeværelse og brugervalg som ovn, vaskemaskine mm. ikke skal indgå i systemet, da brugeren skal foretage disse valg igennem kontakt med underleverandørerne uden om byggelederen.

Sekvensmodellen beskriver de sekvenser det forventes at brugeren skal gennemgå ved brug af registreringssystemet. Sekvensmodellen kan endvidere anvendes som en skabelon til opbygning af registreringssystemet.

1.2 Metode

I dette afsnit er der først opstillet funktionskrav til registreringssystemet, derefter er datamodellen for relationsdatabasen opstillet.

1.2.1 Funktionskrav

Her er funktionskravene til registreringssystemet opstillet, samt uddybet.

Registreringssystemet skal være brugervenligt, overskueligt og let tilgængeligt.

Det er primært over for brugerne at registreringssystemet skal være brugervenligt, da det vil være nødvendigt for en byggeleder at lave ændringer fra projekt til projekt. Brugeren skal bl.a. kunne se de økonomiske konsekvenser af sine valg.

Registreringssystemet skal kunne tilpasses kundernes ønsker, og erfaringsopsamling skal være mulig.

Da løsningsmodellen skal kunne bruges i andre projekter, skal erfaringsopsamling være muligt. Dette vil endvidere skabe et endnu mere brugervenligt registreringssystem ved kommende projekter. Hvis registreringssystemet skal være brugervenligt, skal det være tilpasset kundernes forventninger.

Der må ikke opstå redundante data i registreringssystemet, der skal kun være præcise og relevante data.

Hvis der opstår redundante data i registreringssystemet vil det ikke være overskueligt for byggelederen at redigere og opdatere databasen. Hvis der er flere data end nødvendigt, forsvinder overskueligheden af databasen.

Der skal være mulighed for at sikre registreringssystemet, så oplysninger ikke går tabt ved uheld eller fejl.

Databasen lagres på computer, derfor vil det være muligt at foretage backup, som sikrer registreringssystemet, hvis fejl eller uheld skulle opstå.

Brugerne skal have mulighed for at afprøve ændringer uden at de er bindende.

Registreringssystemet skal udformes således at brugerne har mulighed for at ”lege” med ændringerne uden at de er bindende. Registreringssystemet skal ligeledes give mulighed for at interesserede kan benytte systemet uden at have intention om at købe en bolig.

1.3 Datametoder

Her beskrives hvilke datametoder der er benyttet i miniprojektet, samt sammenhængene mellem dem.

1.3.1 WWW / html

Der er som tidligere beskrevet opbygget et websted i dette miniprojekt. De tilhørende sider er skrevet i HTML via en editor, desuden er der på siderne div. Scripts – specielt ASP. HTML, som er et af de sprog, der kan benyttes til at lave sider, der kan benyttes fra WWW.

Når der opbygges en hjemmeside er det vigtigt, at lave en solid underliggende struktur. Denne filstruktur skal holde styr på html-sider, billeder mv. Det er derved vigtigt at strukturen er nem at overskue og nem at udbygge.

Strukturen på selve hjemmesiden er ligeledes vigtigt. Der skal altid sørges for, at brugeren nemt kan navigere rundt, samt at design/opbygning er tilpasset brugerens niveau. Relevante informationer skal være tilstede, såsom sidste opdatering, hvem der har udført siden osv.

1.3.2 ASP

ASP er et sprog udviklet af Microsoft, som gør det muligt fra WWW at kommunikere med en database. ASP betyder Active Server Pages. ASP kan håndtere andre scripts, såsom JAVA og Visual Basic, hvilket gør at det er muligt at benytte ”almindelige” programmeringsmetoder, såsom løkker, og IF-statements.

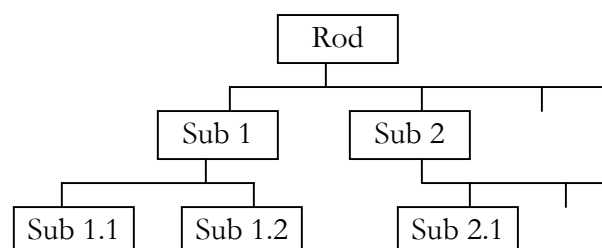
ASP skrives ind i HTML dokumentet, på lige fod med andre scripts. ASP benytter sig bl.a. af følgende objekter.

- Request – Håndterer information sendt fra browseren til serveren (fx. Via cookies) eller fra forms.
- Response - Skriver informationer (fx. HTML koder) fra serveren til browseren.
- Server – Giver adgang til funktioner der håndtere serveren.

1.3.3 Relationsdatabaser og SQL

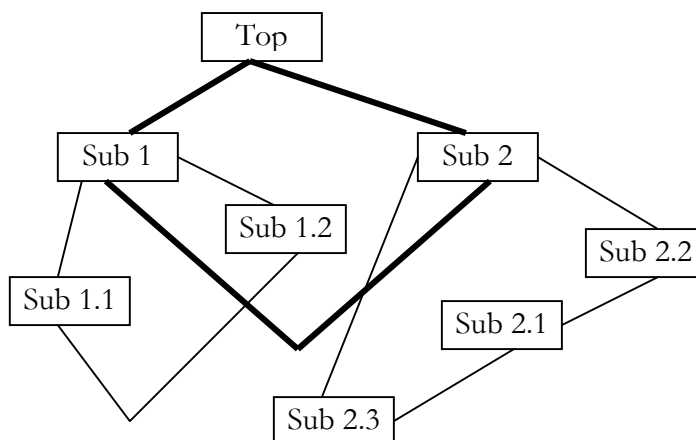
Der findes reelt tre forskellige strukturer på databaser, hierarkisk-, netværks- og relationsdatabaser. Disse tre strukturer gennemgås kort i det følgende.

Hierarkiske databaser er opbygget over ”træer”, dvs. de starter ved roden og bygges op derfra, jf. figur 5. Denne type databaser er hurtige at søge i, men svære at lave ændringer/opdateringer i.



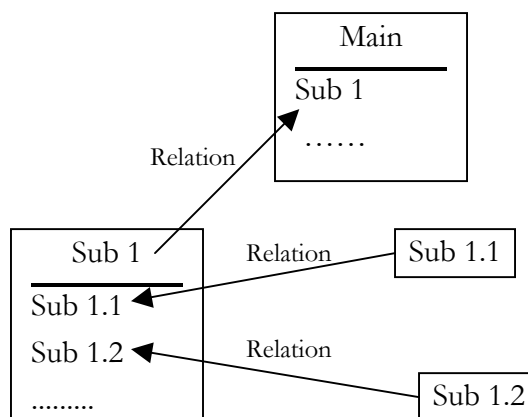
Figur 5. Struktur af hierarkisk database.

Netværksdatabaser er opbygget over associationer over de enkelte enheder, dvs. enhederne er forbundet via referencer, jf. figur 6. Denne struktur gør det væsentligt nemmere at tilføje nye enheder, i forhold til hierarkiske databaser, men strukturen bliver nemt uoverskuelig.



Figur 6. Struktur af netværksdatabase.

Relationsdatabaser er opbygget, som på figur 7. Som ordet relation antyder, er denne form for struktur opbygget ved relationer mellem de enkelte tabeller. Dette giver en simpel struktur, som er nem at vedligeholde og udbygge. Dette skyldes, at der ved relationsdatabaser ikke opstår redundante data, dvs. informationerne står kun ét sted.



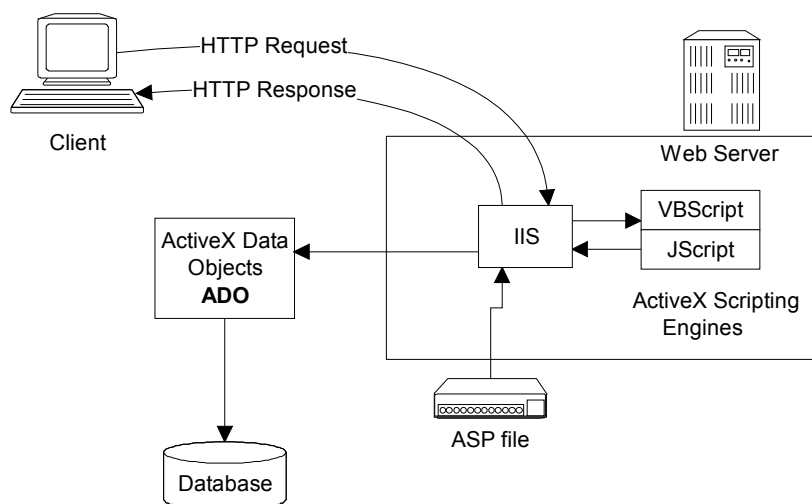
Figur 7. Struktur ved en relationsdatabase.

SQL er et sprog til at kontrollere og anvende relationsdatabaser. SQL er en standard, og står for Structured Query Language.

1.3.4 Sammenhæng mellem database og Klient (via WWW)

Her forklares hvordan en klient kommunikerer med databasen, når dette finder sted via WWW. Dette fremgår af figur 8.

Klienten anmoder webserveren om en side med efternavnet .asp, derved konstaterer serveren, at den skal starte ASP. Hvis det er første gang der sendes en forespørgsel, skal syntaksen kompileres. Dette gøres af Internet Information Server (IIS) der adskiller scripts fra HTML, og afgøre hvilken ”motor” der skal tage sig af div. Scripts. Når kompileringen er færdig, køres scripten, og denne sammenblandes igen med HTML-koderne i asp filen. Den færdige HTML-kode sendes tilbage til brugeren i et http-svar. ADO er et ActiveX-objekt, som sørger for et interface til OLE DB, sådan opnås adgang til stort set enhver database.



Figur 8. Behandling af ASP forespørgsel.

OLE betyder, Object Linking and Embedding. Dvs. at OLE er en Microsoft -standard, der benyttes til at lave objekter i en applikation, så de kan indlægges i et dokument, der er oprettet i en anden applikation. OLE-objektet beholder sit originalformat, og har en kæde til det program, det er oprettet i. Hvis man f.eks. anbringer en tegning - lavet i et tegneprogram - i et tekstdokument, vil man få direkte adgang tegneprogrammets menuer og funktioner, så brugeren kan redigere tegningen direkte i tekstdokumentet, uden at starte tegneprogrammet.

ADO betyder Active Data Objects. Dvs. at ADO er komponenter, der gør det muligt for klientprogrammer at få adgang til - og bearbejde - data i en fil eller serverbaseret database via en udbyder.

1.4 Implementering

I dette afsnit beskrives først hvordan databasen er opbygget, herunder de tabeller som indgår i databasen og relationerne i mellem dem. Dernæst beskrives hjemmesidens opbygning.

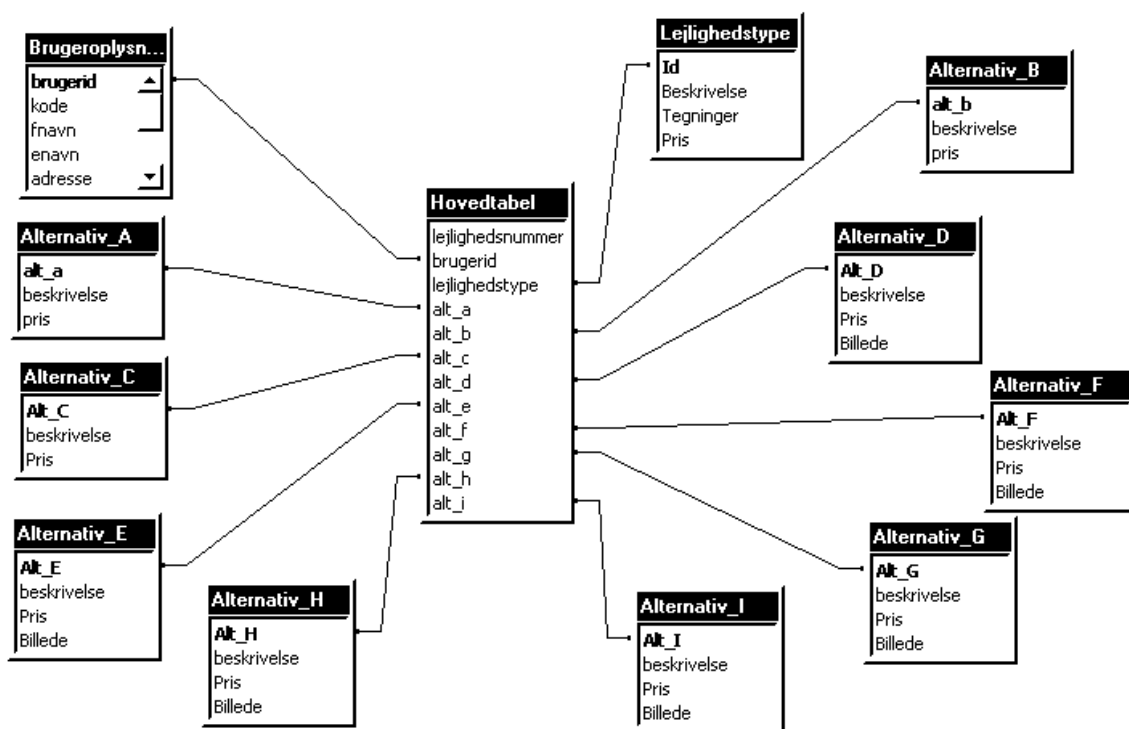
1.4.1 Database

Databasen opbygges i Microsoft Access, i dette program er det muligt at opbygge relationsdatabaser.

Følgende tabeller skal være til stede.

- Tabel med brugeroplysning. Denne tabel skal indeholde data om hver enkel bruger, såsom bruger-ID, password, navn, adresse, tlf.nr., osv.
- Tabeller med de enkelte valgmuligheder. Der laves en tabel for hver enkelt valgttype, indeholdende, valg-ID, beskrivelser, pris og billede.
- Tabel med lejlighedstype, der indeholder de i salgsmaterialet udarbejdede beskrivelser, tegninger og priser.
- Tabeller med leverandørvalg, såsom badeværelse, køkken, mv.
- Hovedtabel indeholdende relationer til de andre tabeller. Det er i denne tabel at en brugers valg bliver registreret, og derved er det over denne tabel at forespørgsler laves.

På figur 9 vises et udsnit af de nævnte tabeller, samt de relationer der er mellem disse og hovedtabellen.



Figur 9. Udsnit af mulige tabeller, samt relationerne imellem.

Det er i selve hovedtabellen at brugerne går ind og tilføjer eller redigerer data. Således undgås det at redundante data opstår.

Databasen opbygges i Microsoft Access ved projektstart. Det er her at byggelederen opretter eller redigerer tabellerne så de er tilpasset det aktuelle projekt. Dvs. at strukturen af databasen laves, og tabeller indeholdende valgmulighederne oprettes. Tabellen med brugeroplysninger, hovedtabellen og tabellerne med leverandørvalg udfyldes løbende via hjemmesiden.

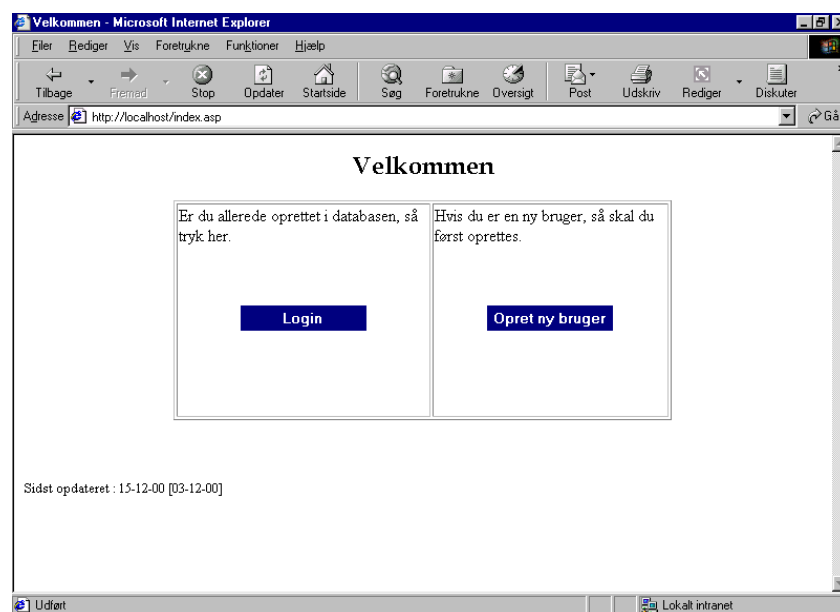
1.4.2 Hjemmeside

Hjemmesiden opbygges først på en lokal computer, hvorefter hjemmesiden er lagt ud på WWW. Hjemmesiden er brugerens mulighed for at foretage valg. I det følgende opstilles de krav som hjemmesiden skal opfylde.

Brugerne skal kunne:

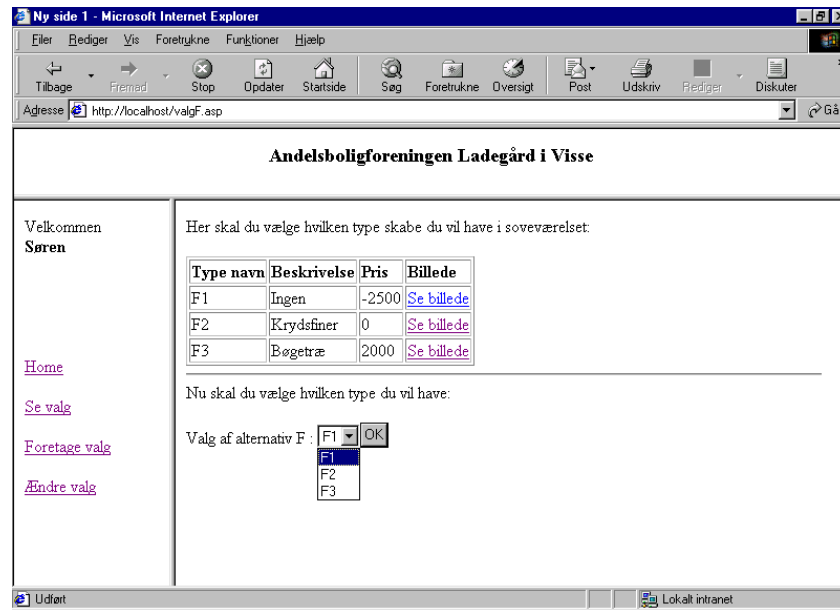
- Oprette sig.
- Foretage valg.
- Ændre valg.
- Se den økonomiske konsekvens af valg.

I det følgende er det vist et uddrag af hvordan brugernes portal ser ud. På figur 10 er vist hvordan brugerne skal identificere sig selv overfor databasen.



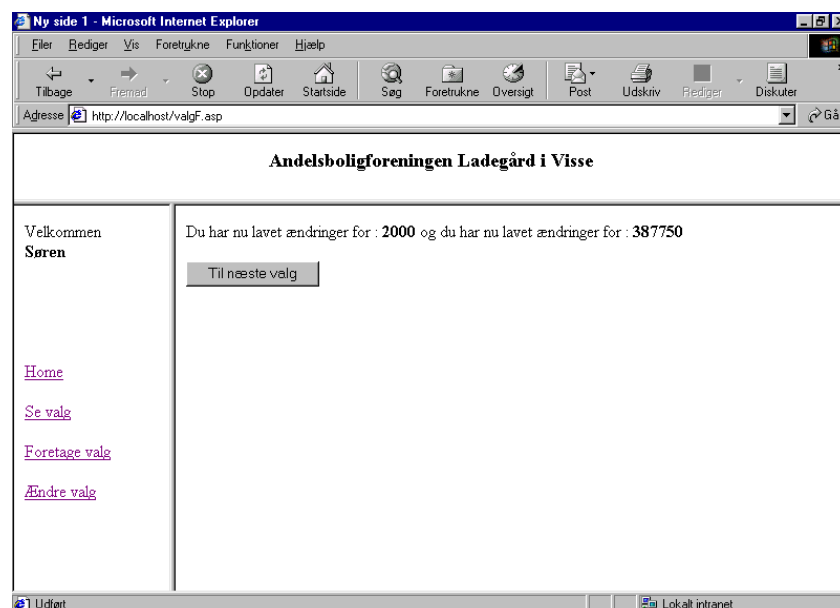
Figur 10. Brugerens identifikation overfor databasen.

Når brugeren har logget ind, bliver han præsenteret for en række valgmuligheder. Når brugeren er nyoprettet, skal han igennem alle valg. Hjemmesiden er derfor opbygget således at valgene sekventielt bliver præsenteret. Når brugeren skal foretage et valg, har han mulighed for at læse en beskrivelse, se prisen samt se et billede af valget, jf. figur 11.



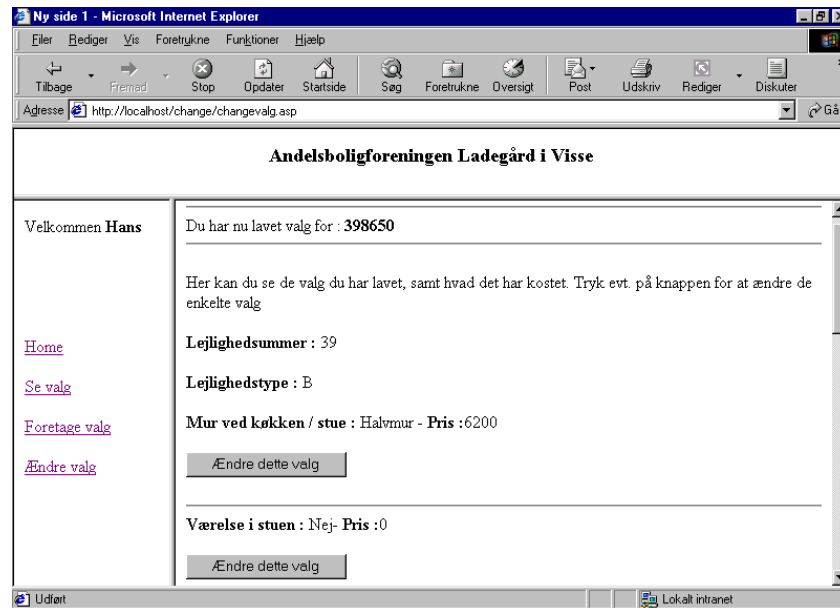
Figur 11. Eksempel på brugerens valgmulighed.

Når brugeren har foretaget sit valg, kan han løbende se de økonomiske konsekvenser, jf. figur 12.



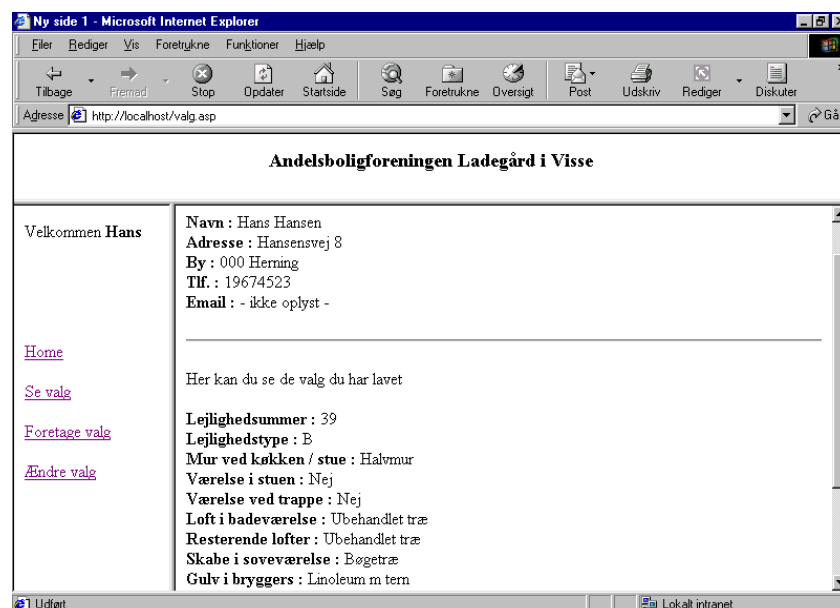
Figur 12. Brugeren kan løbende holde øje med de økonomiske konsekvenser af sine valg.

Når brugeren har foretaget alle valg, vendes tilbage til startside. Her kan bl.a. vælges at ændre sine valg. Her er der mulighed for kun at ændre de valg der ønskes, dvs. ændringsproceduren ikke er sekventielt opbygget, se figur 13.



Figur 13. Efter at have foretaget alle valg, gives der mulighed for at ændre et enkelt valg.

Brugeren kan til enhver tid se de valg der er truffet, se figur 14.



Figur 14. Brugeren kan til en hver tid se sine personlige oplysninger, og de valg han har truffet.

1.5 Vurdering

Registreringssystemet er opbygget som en relationsdatabase, hvor det er opnået at risikoen for at der kommer redundante data er minimeret. Derudover kan man ved brug af relationsdatabasen lave søgninger, og erfaringsopsamling er derved mulig. Ved at hjemmesiden er opbygget som en webside, er det opnået, at adgang til registreringssystemet sker via WWW, og ved at hjemmeside og database

er elektronisk lagret, er der ikke problemer med at tage backup. Brugeren kan ved brug af systemet få mulighed for at foretage valg og se priserne for valgene, dog uden det er bindende.

1.6 Perspektivering

Det udarbejdede registreringssystem styrer de valg brugerne foretager. Dette system virker som et styringsværktøj til ændringsprocessen. Der er dog mange perspektiver i dette styringsværktøj, hvilket der i det følgende redegøres for.

Registreringssystemet kan udvikles, så langt flere valg falder ind under dette. Dvs. brugeren eventuelt kan sammensætte hele sin bolig her. Det kunne herved tænkes, at brugeren direkte kunne forespørge en pris for en given ændring. Denne forespørgsel skulle så bl.a. besvares direkte af fagentreprenøren. Men for at dette skal være muligt, må registreringssystemet indeholde langt flere informationer og være væsentligt bedre til at præsentere dem. Brugere skal have mulighed for at flytte rundt på vægge, skifte farver, indsætte elementer mv. og med det samme kunne se det i en slags 3D model. Det kan tænkes, at der laves et virtuelt hus, som brugeren kan gå rundt i, og foretage alle de forandringer han ønsker, for derefter at købe præcis dette produkt. Dette giver brugeren en enestående mulighed for at se konsekvenserne af sine valg, både mht. udseende og økonomi.

Registreringssystemet kunne også udbygges til at være et generelt styringsværktøj på en byggesag. Det tænkes, at registreringssystemet kunne lagre alle informationer, såsom tegninger, mødereferater, regnskaber, tidsplaner osv. Registreringssystemet er på nuværende tidspunkt tænkt som et system til registrering af valg, men fungerer også som et informationssystem for byggelederen. Hvis systemet udvides til at omhandle hele byggesagen, ville det derved både være et informations- og registreringssystem, som alle parter kunne benytte. Herved skabes et styringsredskab, der er centralt placeret i byggesagen og tilgængelig for alle parter.

Idet registreringssystemet skaber en mere effektiv kommunikation, sørger det for, at der sker færre fejl i byggeprocessen. Hvis der sker færre fejl, bliver kvaliteten af byggeriet sandsynligvis forbedret. Systemet kunne tænkes udbygget, så det tager endnu mere hensyn til de kvalitetsmæssige aspekter i byggesagen.

Referenceliste

- | | |
|---|---|
| Access 2000 | Access 2000 Håndbogen 1
Celeste Robinson
IDG
ISBN: 87-7843-319-3 |
| Access 2000 | Access 2000 Håndbogen 2
Celeste Robinson
IDG
ISBN: 87-7843-320-7 |
| ASP | ASP – Active Server Pages
Wille Koller
IDG
ISBN: 87-7843-361-4 |
| Diversity project | User Requirements Formulations and human Computer Interface
Design in the Diversity project. Context and methods. Initial study.
Per Christiansson, m.fl.
Aalborg 17.12.2000 |
| http://it.civil.auc.dk | http://it.civil.auc.dk
Dato: 10/1 – 2001 |
| http://www.it-leksikon.dk | http://www.it-leksikon.dk
Dato: 10/1 – 2001 |
| SQL | Start på SQL
F.D. Rolland
IDG
ISBN: 87-7843-169-7 |

Appendiks 1: Udskrifter fra koderne, ASP og HTML

I dette appendiks er der udskrifter fra hjemmesiderne, indeholdende HTML- og ASP-koder. Der udvalgt nogle koder, så de mest gængse strukturer og metoder fremkommer.

Følgende er fra filen opretnybruger5.asp, der opretter en ny bruger i databasen, med de ønskede oplysninger

```
<% @language = VBScript %>
<%
    Option Explicit
    Response.Expires = 0

    dim strBrugerID, strKode, strFnavn, strEnavn, strAdresse, strPostnr, strTown,
    strTlf, strEmail
    dim objConn, objRS, strQ
    dim strConnection, strOutput

    if Request.ServerVariables("CONTENT_LENGTH") <> 0 Then

        set objConn = Server.CreateObject("ADODB.connection")
        '----- opretter forbindelse og åbner databasen-----
        strConnection = "DSN=IT1;Database=IT1"
        StrConnection = Strconnection & "UID=;PWD=;"
        objConn.Open strConnection

        ' -----Modtager formdata fra body -----
        strBrugerID = Trim(Request.Form("brugerID"))
        strkode = Trim(Request.Form("kode"))
        strFnavn = Trim(Request.Form("fnavn"))
        strEnavn = Trim(Request.Form("enavn"))
        strAdresse = Trim(Request.Form("adresse"))
        strPostnr = Trim(Request.Form("postnr"))
        strTown = Trim(Request.Form("town"))
        strEmail = Trim(Request.Form("email"))
        strTlf = Trim(Request.Form("tlf"))

        ' ----- Laver SQL sætninger -----
        strQ = "INSERT INTO brugeroplysning (brugerid, kode, fnavn, enavn, adresse,
        postnummer, tlf, email, town) VALUES "

        strQ = strQ & "(" & strbrugerid & "," & strKode & "," & strFnavn & "," & strEnavn &
        "," & strAdresse & "," & strPostnr & "," & strTlf & "," & strEmail & "," & strtown &
        ")"

        Set objRS = objConn.Execute(strQ)

        strQ = "INSERT INTO hovedtabel (brugerid) VALUES "
        strQ = strQ & "(" & strbrugerid & ")"
```

```
Set objRS = objConn.Execute(strQ)
```

```
Response.Cookies("ITProjekt")= Request.Form("BrugerID")
Response.Cookies("ITProjekt").Expires="Jan 1, 2010"
%>
```

```
<html><body>
```

```
Du har indtastet følgende data : <br><br>
<b>Navn : </b> <%= StrFnavn %> <%= StrEnavn %> <br>
<b>Adresse : </B> <%= StrAdresse %> <br>
<b>By : </B> <%= StrPostnr %> <%= StrTown %> <br>
<b>Tlf.: </B> <%= StrTlf %> <br>
<b>Email : </B> <%= StrEmail %> <br>
<br>
<b>Bruger ID : </B> <%= StrBrugerID %> <br>
<b>Kode : </B> <%= StrKode %>
<br><br>
```

Tryk her for at komme inde i databasen.

```
<br>
<FORM ACTION="/opstart.asp" METHOD="POST">
<input type="Submit" Name="submit" Value="Til hovedmenuen">
```

```
<%
objRS.close
objConn.close
Set objRS = Nothing
Set objConn = Nothing
%>
```

```
    </body></html>
<% else %>
```

```
<html><body>
```

```
    Indtast ønskede oplysninger: <br><br>
<form action="/opretnybruger5.asp" method="post">
Bruger ID : <input type="text" name="brugerID" size="7"><BR>
Adgangskode : <input type="text" name="kode" size="7"><BR>
Fornavn : <input type="text" name="fnavn" size="12"><BR>
Efternavn : <input type="text" name="enavn" size="12"><BR>
Adresse : <input type="text" name="adresse"><BR>
Postnummer : <input type="text" name="postnr" size="7"><BR>
By : <input type="text" name="town"><BR>
Telefonnummer : <input type="text" name="tlf"><BR>
Email : <input type="text" name="email"><BR>
<input type="submit" name="submit" value="Opret bruger">
</form>
<br><br><br></body>
</html>
```

```
<% End If %>
```

Det følgende er fra filen valgalteu.asp, der præsenterer brugeren for de muligheder der ved en valgtype, og derefter gemmer disse informationer i database.

```
<% @language = VBScript %>
<%
  Option Explicit
  Response.Expires = 0

  Dim strFirstname
  Dim objConn, objRS, strQ, strbrugerid
  Dim strConnection, stroutput, stralte
  Dim objRS1, objRS2, objRS3, objRS4, objRS5

  ' ----- opretter forbindelse og åbner databasen
  Set objConn = Server.CreateObject("ADODB.connection")
  StrConnection = "DSN=IT1;Database=IT1"
  StrConnection = StrConnection
  objConn.Open strConnection

  If Request.ServerVariables("CONTENT_LENGTH") <> 0 Then

    stralte = TRIM(Request.Form("alte"))
    strbrugerid = Request.Cookies("ITProjekt")
    strQ = "UPDATE hovedtabel SET alt_e = " & stralte & " WHERE brugerid = " & strbrugerid &
    """"

    Set objRS = objConn.Execute(strQ)

  %>
  <html><body>
  <%
    strQ = "SELECT hovedtabel.brugerid, Alternativ_E.Pris "
    strQ = strQ & " FROM Alternativ_E INNER JOIN hovedtabel ON Alternativ_E.Alt_E =
    hovedtabel.alt_e"
    strQ = strQ & " WHERE hovedtabel.brugerid LIKE "
    strQ = strQ & """" & strbrugerid & """"
    Set objRS = objConn.Execute(StrQ)
  %>
  Du har nu lavet ændringer for :<b> <%= objRS("pris") %></B>
  <%
    strQ = "SELECT hovedtabel.brugerid, Alternativ_D.Pris "
    strQ = strQ & " FROM Alternativ_D INNER JOIN hovedtabel ON Alternativ_D.Alt_D =
    hovedtabel.alt_d"
    strQ = strQ & " WHERE hovedtabel.brugerid LIKE "
    strQ = strQ & """" & strbrugerid & """"
```

```
Set objRS1 = objConn.Execute(StrQ)
```

```
strQ = "SELECT hovedtabel.brugered, Alternativ_C.Pris "
strQ = strQ & " FROM Alternativ_C INNER JOIN hovedtabel ON Alternativ_C.Alt_C =
hovedtabel.alt_c"
strQ = strQ & " WHERE hovedtabel.brugered LIKE "
strQ = strQ & """" & strbrugered & """"
Set objRS2 = objConn.Execute(StrQ)
```

```
strQ = "SELECT hovedtabel.brugered, Alternativ_B.Pris "
strQ = strQ & " FROM Alternativ_B INNER JOIN hovedtabel ON Alternativ_B.Alt_B =
hovedtabel.alt_b"
strQ = strQ & " WHERE hovedtabel.brugered LIKE "
strQ = strQ & """" & strbrugered & """"
Set objRS3 = objConn.Execute(StrQ)
```

```
strQ = "SELECT hovedtabel.brugered, Lejlighedstype.Pris "
strQ = strQ & " FROM Lejlighedstype INNER JOIN hovedtabel ON Lejlighedstype.Id =
hovedtabel.lejlighedstype "
strQ = strQ & " WHERE hovedtabel.brugered LIKE "
strQ = strQ & """" & strbrugered & """"
Set objRS4 = objConn.Execute(StrQ)
```

```
strQ = "SELECT hovedtabel.brugered, alternativ_a.Pris "
strQ = strQ & " FROM alternativ_a INNER JOIN hovedtabel ON alternativ_a.alt_a =
hovedtabel.alt_a "
strQ = strQ & " WHERE hovedtabel.brugered LIKE "
strQ = strQ & """" & strbrugered & """"
Set objRS5 = objConn.Execute(StrQ)
%>
```

og du har nu lavet ændringer for :

```
<%=objRS("pris")+objRS1("pris")+objRS2("pris")+objRS3("pris")+objRS4("pris")+objRS5("pris")
%>
```

```
<br><br>
```

```
<FORM ACTION="/valgF.asp" target="_parent" METHOD="POST">
```

```
<input type="Submit" Name="submit" Value="Til næste valg">
```

```
<% else %>
```

```
</form>
```

Her skal du vælge hvilken type loft du vil have i de resterende rum:


```
<table border="1">
```

```
<tr>
```

```
<td>
```

```
<B>Type navn</B>
```

```
</TD>
```

```
<td>
```

```
<B>Beskrivelse</B>
```

```
</TD>
```

```
<td>
```

```

<b>Pris</B>
<td>
<b>Billede</B>
</TD>
</TD>
</TR>
<%
' ----- SQL søgningskriterer laves
StrQ = "SELECT Alternativ_E.Alt_E, Alternativ_E.Beskrivelse, Alternativ_E.Pris,
Alternativ_E.Billede "
StrQ = StrQ & " FROM Alternativ_E "
Set objRS = objConn.Execute(StrQ)

while not objRS.EOF
Response.write "<tr>"
Response.write "<td>"
Response.write objRS("Alt_E") & " "
Response.write "</TD>"
Response.write "<td>"
Response.write objRS("Beskrivelse") & " "
Response.write "</TD>"
Response.write "<td>"
Response.write objRS("Pris") & " "
Response.write "</TD>"
Response.write "<td>%">
<A href="http://localhost/pictures/
<% Response.write objRS("billede")
%>
" target="loft1"> Se billede </A>
<%
Response.write "</TD>"
Response.write "</TR>"
objRS.MoveNext
WEND
Response.write "</table>"

objRS.close
objConn.close
Set objRS = Nothing
Set objConn = Nothing

%>

<hr>
Nu skal du vælge hvilken type du vil have: <BR>
<FORM ACTION="/valgalteu.asp" METHOD="POST">
Valg af alternativ E :
<select size="1" name="alte">
<option>E1</option>

```

```
<option>E2</option>
<option>E3</option>
</select><input type="submit" value="OK" name="submit">
<BR>
<BR>

</FORM></body>
<% End if %>
```