# Analyse

Opgaveformuleringen: Krav specifikationen fra Sydvest-Bo fortæller om nogle ret konkrete relationer som eksister imellem de informationer, data firmaet håndterer:

* En række personer
* der optræder i forskellige sammenhænge
* og udfylder forskellige roller og gøremål, resulterende i at der produceres
* 4 konkrete kontrakt typer
* og noget kommunikation
* og noget bruger interaktivitet
* og lidt restriktioner.

Vores Database design kæder relationerne sammen på en konsistent måde, men det efterlader ret åbenlyse huller. Hertil introducerer vi nogle views i databasen:

* En udlejningstabel, hvoraf det fremgår hvornår noget er ledigt
* Opgave lister til medarbejdere

Der stilles også krav om ….

De enkelte konsulenter, kunder, udlejere og handy-mænd skal på forskellig vis kunne navigere rundt og betjene programmet, så de intuitivt får nogle relevante informationer og andre måske hemmelig holdes.

Derfor er det ønskelig med en GUI der kan levere dette og f.eks. kan identificere de enkelte brugere vha. et login.

# DB

Vores database (db) design afspejler sammenhængene skitseret ovenfor. Der er lag vægt på at opdele tabellerne i forhold til de relationer der optræder i kravspecifikationen og der er gjort et forsøg på at udskille redundante data i mindre tabeller. Intuitive sammenkog fra databasen er lagt over views, og uintuitive sammenhænge er lagt over i metoder der laver mere komplekse forespørgsler til databasen. Dette skyldes at disse metoder skal kunne arbejde på en opdaterede data fra basen, og ikke et view der blev auto genereret tidligere. Det kunne f.eks. være at en udlejning netop har fundet sted og at en anden kunde samtidig prøver at leje den samme hybel samtidig. Et en ad gangen tjek, efter først til mølle princippet.

Krav specifikationen rummer mulighed for en rationalisering af database designet ved at lade alle ferieresidenser, deres ejere og opsynsmænd bo i fælles klasser. Disse respektive tabeller tilføjes et type felt til arts bestemmelse.

Billede tekst: Ønsker de også tydeligere attributter?

Det eneste der egentlige stikker ud, er introduktionen af et password. Denne attribut der sidder på enhver person. Af flere årsager, men bl.a. som udslag af et køns-kompromis, har hver person altså, uanset køn, netop en disse attributter.

Af tabellernes design følger en række intuitive klasser.

# Intuitive klasser

De intuitive klasser udgøres primært som en parallelitet til databasens tabeller. Ved at sammensættes og arve fra hinanden og skal disse tabel-klasser kunne opbygge, rumme, opdatere og slette data fra databasen. De skal rumme metoder til at sammenstrikke hele sql sætninger eller i det mindste få jobbet udført.

Parallelt skal disse klasser kunne indgå i et lignende regi til at afgive bruger I/O: indhold til skærmen, en kontrakt, en email. Overordnet skal output fra nedarvede klasser og metoder kunne returnere små formaterede strenge. Disse skal igen af en overordnet klasse kunne sammenflettes til linjer af formateret tekst, der igen kan afleveres til noget: GUI, email eller db’en, som håndtere og laver det endelige output format.

# Gui

For at udnytte det maximale af konsolskærmen benytter vi et par biblioteker til at maksimere skærmens areal:

using System.Runtime.InteropServices;

using System.Text;

using System.IO;

Herved få vi også fastlagt skærmen størrelse på brugerens computer og kan fastslå de eksakte koordinaters og mål til at lave brugerfladen: et centrum, en ønskelig pop-op vindues størrelse og lokation, placeringer af et vindues heiraki/stak, proportionerne imellem vinduesbufferen og skærmstørrelsen, mm.

## Objekt vindue

Vi laver et objekt pop-op-vindue der kan drage nytte af disse egenskaber.  
Objekt Property x og y fortæller hvor vinduet starter: ca. centrum -½ skærm udstrækning og lidt fra... Objekt Property w og h (width og height) er udstrækningen af pop-op-vinduet: ca. 2 \* (centrum -½ skærm udstrækning og lidt fra).   
Vi kan ombryde tekst i vinduet, da vi nu kender vinduets udstrækningen. Varierede metoder laves til at udskrive tekstbeskeder, menuer, Tabel indhold, placering af tekst input.  
Dette fordre nogle igen forskellige typer af vinduer: menuer, pop-op, tekst input, scrolle til sager der ikke kan være på skærmen. (Her satses benhårdt på at de uattraktive ferieuger ikke automatisk er i fokus)  
Når der trækkes lidt fra af hensyn li layoutet, marginer, skygger og placering på skærmen.

## Objekt skærm

Uden om vinduet kan vi bygge et objekt skærm, der holder en stak eller liste af vindues instanser. Elementerne kan tilføjes, byttes rundt eller fjernes. Gennem løbes stakken tegnes gentegnes skærmen i den ønskede rækkefølge.

## Objekt skærmrække

Ved at pakke objekt skærme ind i et nyt objekt som ovenfor, kan vi facilitere at flere skærme koeksistere men kun en enkelt vises ad gangen.

# Mindre intuitive klasser og metoder