



Handin 1 - I4DAB

GRUPPE 30

Simon Møller Hindkær - 201609970 (au570382)

Christoffer Kjellerup Jakobsen - 201610457 (au569759)

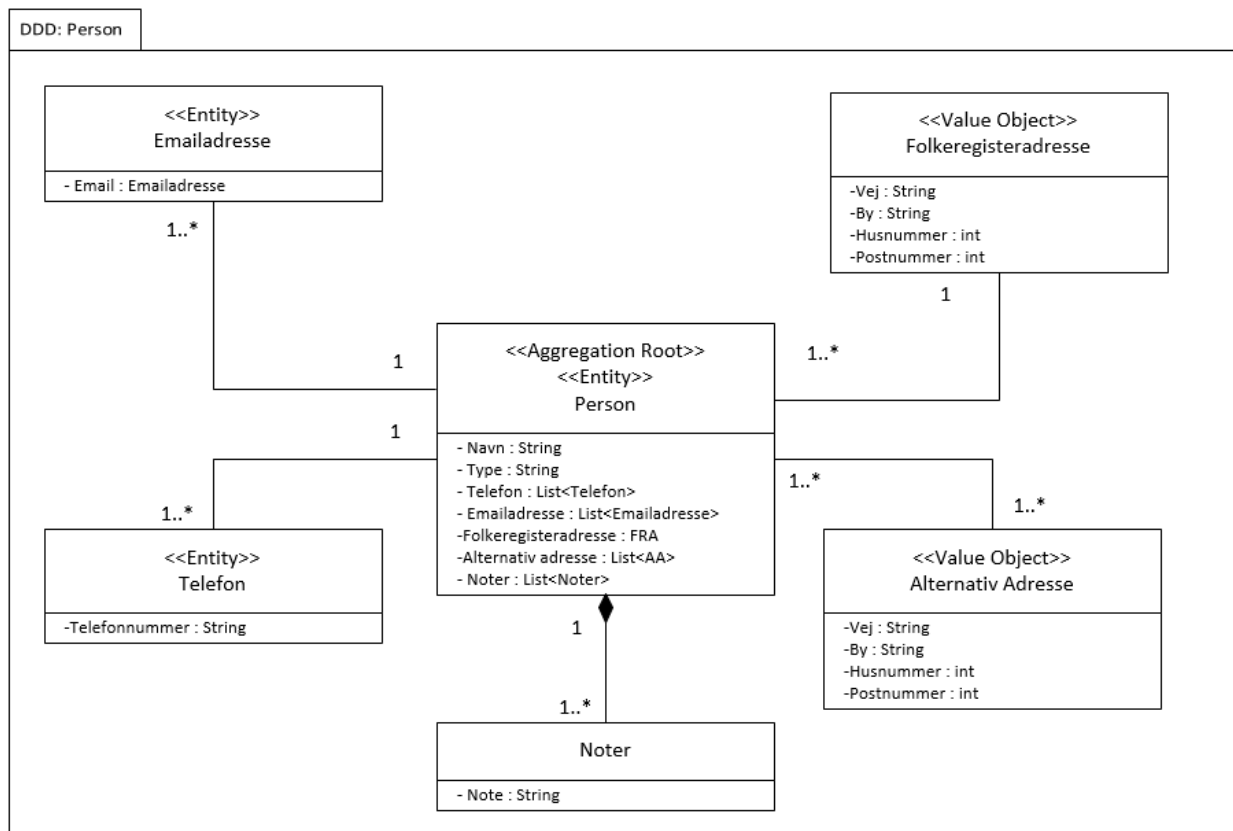
Indhold

Indledning	2
1. Domain Driven Design i UML	2
2. JSON + XML for DDD	3
Eksempel af JSON for klassen "Person"	3
Objektillustration for testperson "Janus"	3
Eksempel for XML for klassen "Person"	4
3. Entity Relationship Diagram (ERD)	5
ERD	5
DSD	6
4. Scaffolding og graf-træ model	7
Scaffolding model	7
Graf-træ model	8
5. Sekvensdiagram over database	8

Indledning

Vi vil i denne journal opbygge den første udgave af en applikationsarkitektur, for applikationen "Personkartotek", ved hjælp af domænemodellering. Journalen vil indeholde et Domain Driven Design (DDD) i UML, samt JSON og XML model af her omtalte domænemodel. Herudover vil vi udarbejde et Entity Relationship Diagram (ERD), samt et scaffolding diagram og Graf-trædiagram.

1. Domain Driven Design i UML



I vores domain driven design kan vi se at vores aggregation root er person. Dette skyldes, at det er det eneste "hele" objekt, som der bliver hentet ned. Denne root indeholder så en række value objects, som bliver udgjort af en E-mail, en adresse og en telefon. Disse små valueobjects, repræsenterer simple entiteter.

2. JSON + XML for DDD

I dette afsnit vil der blive vist JSON og XML eksempler for vores DDD.

Eksempel af JSON for klassen "Person"

```
1 {
2   "Type": "Student",
3   "Navn": "Janus Phillip Jørgen Jatak",
4   "Telefon": [
5     {
6       "Telefontype": "Mobil",
7       "Teleselskab": "3",
8       "Telefonnummer": 88888888
9     }
10  ],
11  "Email_Adresse": [
12    {
13      "Adresse": "Janus.P.J.Jatak@gmail.com"
14    }
15  ],
16
17  "Noter": [
18    {
19      "Note": "Han er en god fyr."
20    }
21  ],
22  "Adresse": [
23    {
24      "Adresstype": "Folkeregisteradresse",
25      "Vejnavn": "Morgenstjernevej",
26      "Husnummer": "40",
27      "Postnummer": 9900,
28      "Bynavn": "Nødsaxe"
29    }
30  ],
31  "PersonID": "303044-0000"
32 }
33
```

Objektillustration for testperson "Janus"

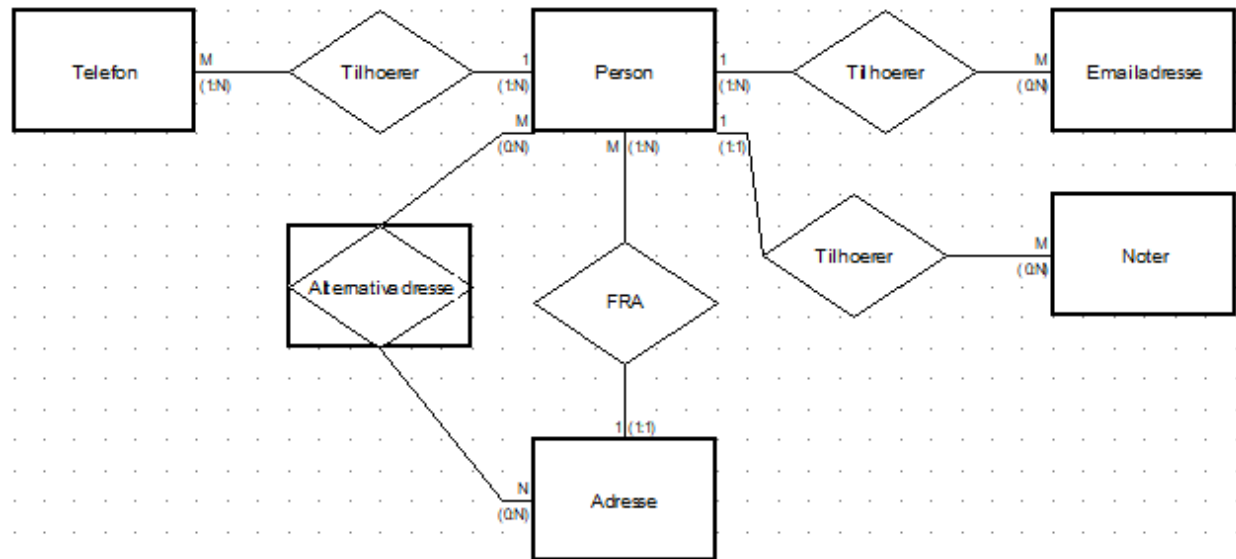
```
object {7}
  Type : Student
  Navn : Janus Phillip Jørgen Jatak
  Telefon [1]
    0 {3}
      Telefontype : Mobil
      Teleselskab : 3
      Telefonnummer : 88888888
  Email_Adresse [1]
    0 {1}
      Adresse : Janus.P.J.Jatak@gmail.com
  Adresse [1]
    0 {5}
      Adresstype : Folkeregisteradresse
      Vejnavn : Morgenstjernevej
      Husnummer : 40
      Postnummer : 9900
      Bynavn : Nødsaxe
  PersonID : 303044-0000
  Noter [1]
    0 {1}
      Note : Han er en god fyr.
```

Eksempel for XML for klassen "Person"

```
1  <?xml version="1.0"?>
2  <Type> Student</Type>
3  <Person>
4      <Fornavn>Janus</Fornavn>
5      <Mellemlnavn>Jørgen</Mellemlnavn>
6      <Efternavn>Jatak</Efternavn>
7  </Person>
8
9  <Telefon>
10     <Telefontype>Mobil</Telefontype>
11     <Teleselskab>3</Teleselskab>
12     <Telefonnummer>83838383</Telefonnummer>
13 </Telefon>
14
15 <Email_Adresse>
16     <Adresse>Janus.P.J.Jatak@gmail.com</Adresse>
17 </Email_Adresse>
18
19 <Noter>
20     <Note>Han er en god fyr.</Note>
21 </Noter>
22
23 <Adresse>
24     <Adresstype>Folkeregisteradresse</Adresstype>
25     <Vejnavn>Morgenstjernevej</Vejnavn>
26     <Husnummer>40</Husnummer>
27     <Postnummer>9900</Postnummer>
28     <Bynavn>Nødsaxe</Bynavn>
29 </Adresse>
30 <PersonID>303044-0000</PersonID>
```

3. Entity Relationship Diagram (ERD)

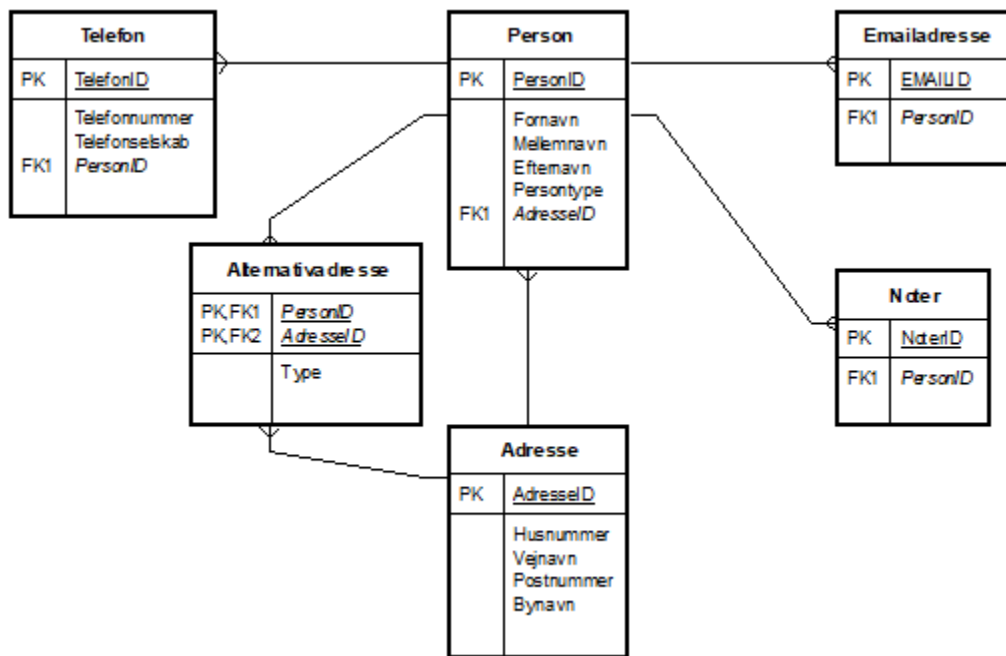
ERD



En person SKAL have et eller flere telefonnumre
En person KAN have en eller flere E-mail adresser
En person SKAL have en Folkeregister adresse
En person KAN have en Alternativ adresse
En person KAN have en til flere Noter

ERD modellen bruges til at få et visuelt billede af relationerne i databasen. Vi kan se at vi i ERD modellen har 2 binary degrees mellem Telefon og Person, og Person og Adresse. Derudover har vi forskellige connectivities mellem vores entities. Et eksempel kan være, at en Person kan have flere alternative adresser, og der er kan flere personer som bor på sådan en adresse, så derfor er den connectivity sat som many to many. Alle relationerne i vores ERD er mandatory jvf. Opgavebeskrivelsen.

DSD

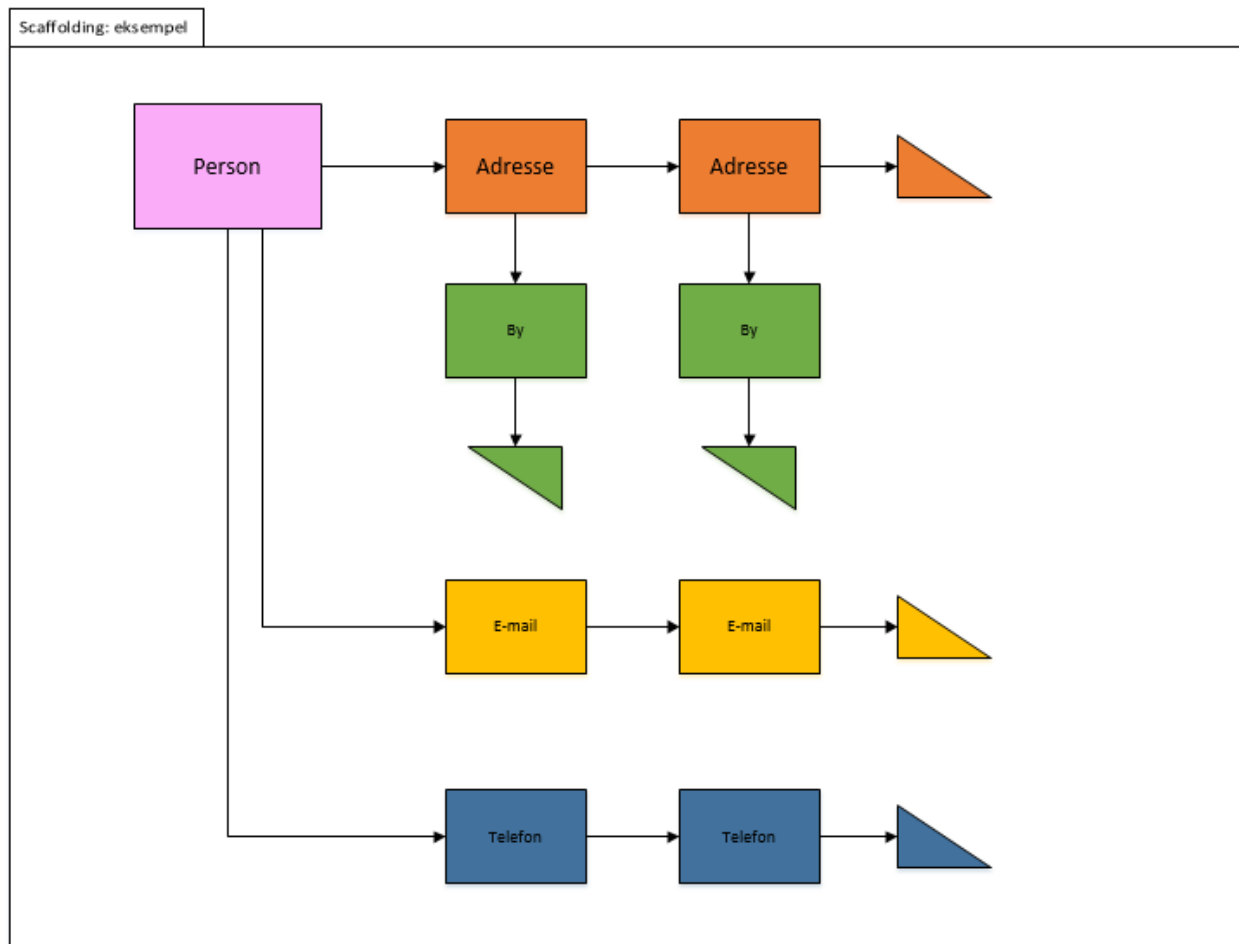


Vores DSD giver os et overblik over de forskellige attributter de forskellige entiteter har. Herudover kan man bl.a. også se, at den alternative adresse kan være af forskellige type.

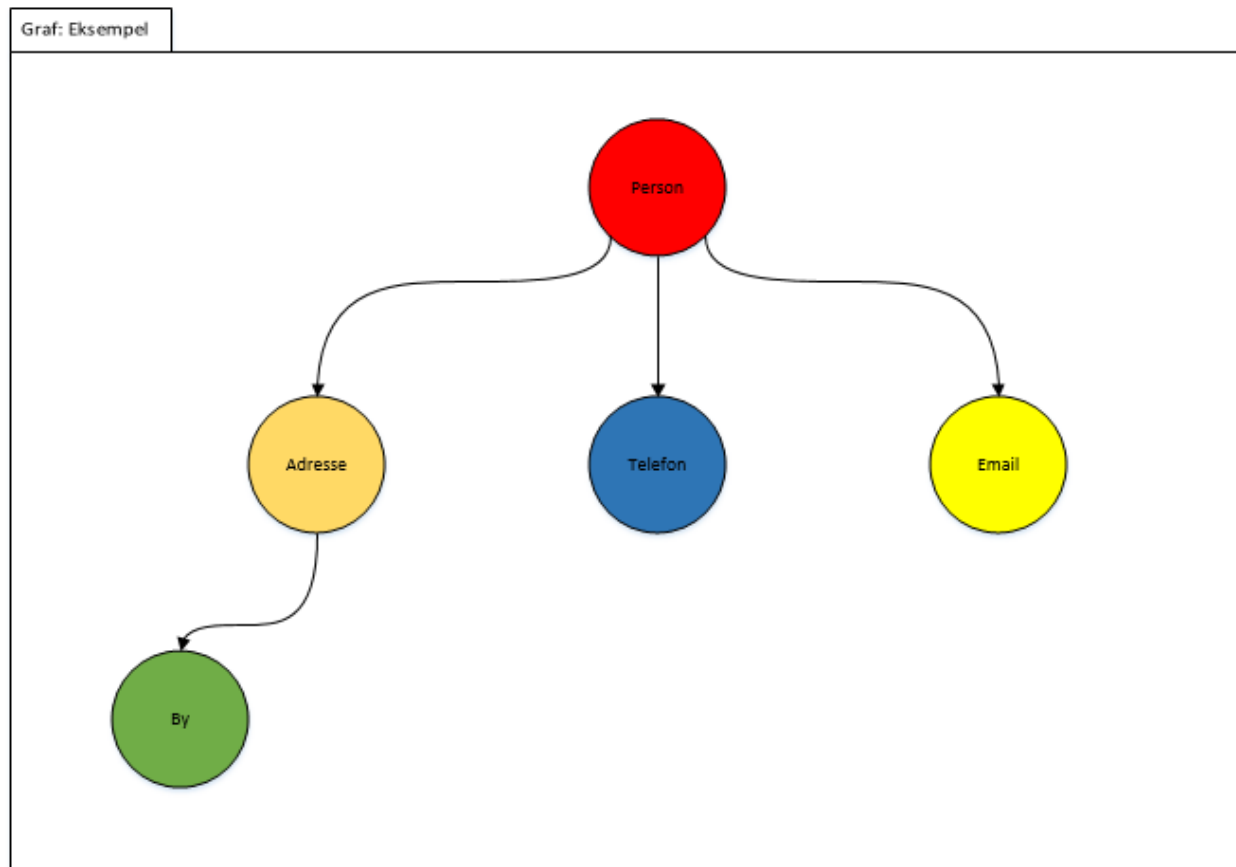
4. Scaffolding og graf-træ model

Ud fra et eksempel med en testperson, har vi udarbejdet en scaffolding model, samt en graf-træ's model. Dette er lavet for at give os et indblik i hvordan de forskellige entiteter snakker sammen, i vores system.

Scaffolding model



Graf-træ model



5. Sekvensdiagram over database

For at give et bedre overblik over applikationen, har vi udarbejdet et sekvensdiagram, for et scenarie, hvor vi vil have sat en bestemt persons adresse.

