

# I4DAB - Handin 2

## Studienummer

201610981

## Indhold

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduktion</b>                         | <b>2</b> |
| <b>2</b> | <b>ERD &amp; DDD</b>                        | <b>2</b> |
| 2.1      | DDD - Domain Driven Design . . . . .        | 2        |
| <b>3</b> | <b>User stories</b>                         | <b>3</b> |
| 3.1      | ERD - Entity Relationship Diagram . . . . . | 4        |
| <b>4</b> | <b>JSON &amp; XML</b>                       | <b>5</b> |
| 4.1      | JSON . . . . .                              | 5        |
| 4.2      | XML . . . . .                               | 6        |
| <b>5</b> | <b>Overordnet SW-Arkitektur</b>             | <b>7</b> |
| <b>6</b> | <b>Scaffolding-diagram &amp; Graf</b>       | <b>8</b> |
| 6.1      | Scaffolding-diagram . . . . .               | 8        |
| 6.2      | Graph . . . . .                             | 9        |

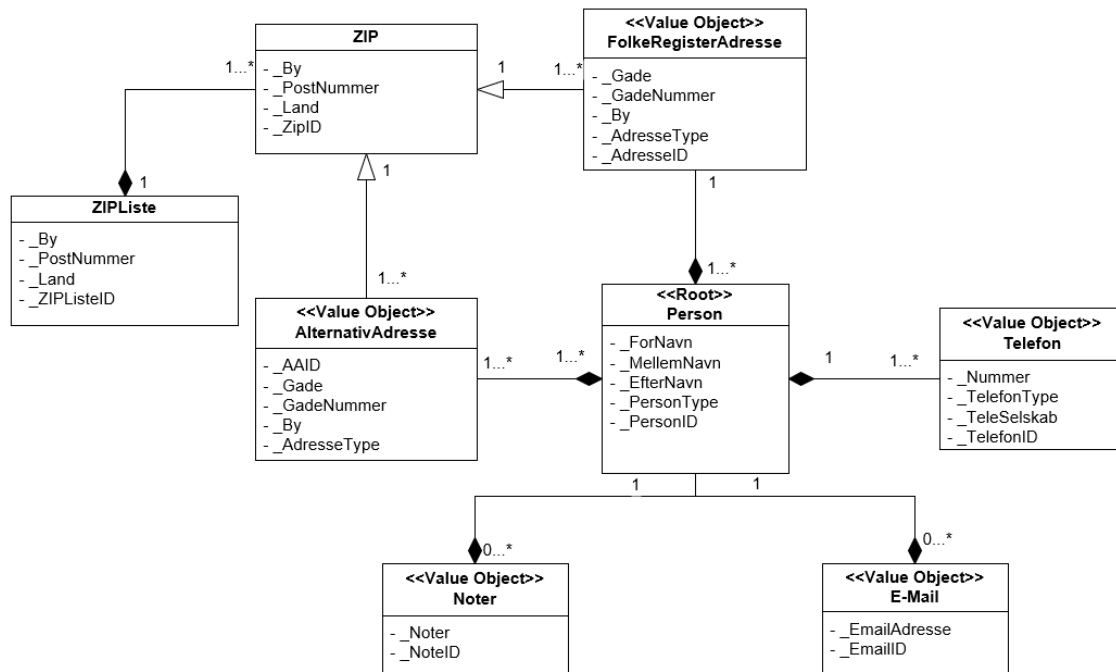
# 1 Introduktion

Denne Handin er en fortsættelse af Handin 1 i I4DAB. Denne her gang skal man individuelt implementere Personkartoteket, hvori der også skal implementeres en database.

## 2 ERD & DDD

### 2.1 DDD - Domain Driven Design

Ud fra oplysningerne der er givet i opgaveformuleringen er der blevet lavet en DDD (Domain Driven Design)



Figur 1: Domain Driven Design

**Aggregate:** Der anvendes Aggregate da der er komplekse associationer. Vi har en række objekter der ses som en enhed, der kaldes Aggregate. Objekterne Telefon, Email, Adresse og Person udgør aggregate. Når der sker en dataændring hos en af objekterne, har det en betydning for de andre tilknyttede objekter.

**Root:** Person er blevet valgt som Root, da det er den entity der indebærer en identitet. Det der karakteriserer identiteten for objektet, Person, er de to følgende eksempler:

- Attribut af objektet, Personkartotek.
- En kombination af attributterne.
  - Telefon
  - Adresse
  - Email

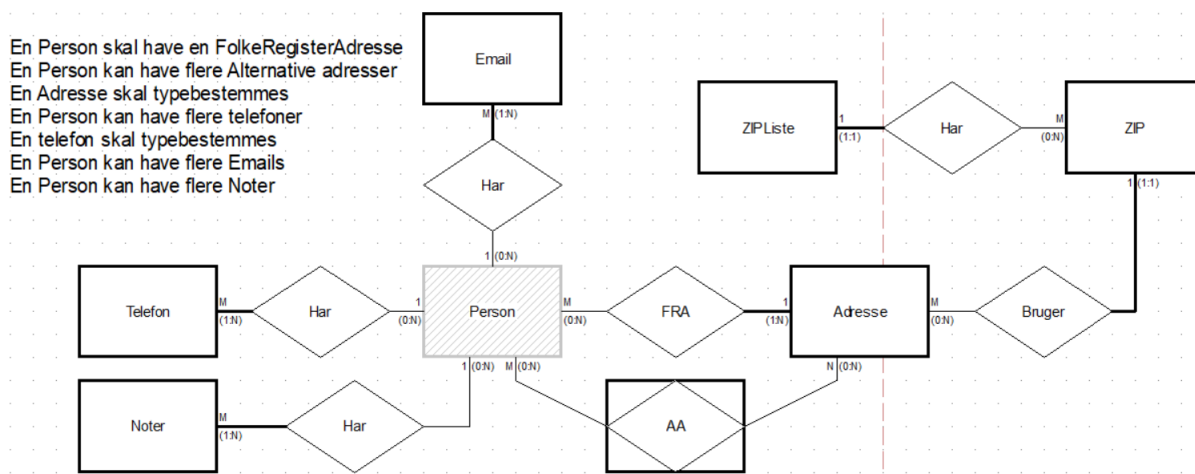
**Entity:** Entity er en vigtig del af et domæne model. Et objekt med en identitet bliver kaldt for en "Entity". Dette kan ses i opgaven hvor objektet, Person, har en identitet.

**Value object:** Et value object bør altid tilhøre en eller flere enheder, det kan ikke leve alene. Dette ses fx med Adresse, Telefon og E-mail.

### 3 User stories

- Som bruger, kan jeg sende en mail  
Til en Person i Personkartoteket  
Så jeg kan kontakte Personen via mail.
- Som bruger, kan jeg ringe  
Til en Person i Personkartotek  
Så jeg kan kommunikerer med Personen.
- Som bruger, kan jeg møde en Person  
Fra Personkartoteket  
Så jeg kan få fysisk kontakt til Personen.
- Som bruger, kan jeg slette en Person  
Fra Personkartoteket  
Idet, at Person ikke længere er i Personkartoteket.

### 3.1 ERD - Entity Relationship Diagram



Figur 2: Entity Relationship Design

**Kardinalitet:** Kardinalitet fortæller noget om, hvor mange af hver entitet der indgår i en relation. Kardinalitet udtrykker den mindste og største antal af entitet forekomster, som er associeret med en forekomst af den relateret entitet.

**Connectivity:** Begrebet "Connectivity" bliver brugt til at beskrive klassifikationen i relationerne. Det gør man for at vise eksistensen af relationerne og typen af relation mellem entiteterne. Det hedder også relations-type:

- 1-til-1
- 1-til-mange
- mange-til-mange

På ERD-diagrammet (Figur 2) kan man se en 1-til-mange relation mellem entiteterne, Person og Adresse.

**Degree:** Degree er antallet af entiteter, som er involveret i forholdet og i vores tilfælde er den 2 (Binary relationship). Forholdet kunne dog også være "unary" og "ternary".

**Deltagelse:** Deltagelse beskriver en relationship, hvor mindst en record i en tabel skal eksisterer, før der kan være flere records under samme tabel.

**Navigering :** Vi finder objekter ved at følge reference af andre objekter.

## 4 JSON & XML

### 4.1 JSON

```
1 {
2   "Navn": {
3     "ForNavn": "Danny",
4     "MellemNavn": "something",
5     "EfterNavn": "Tran",
6   },
7   "PersonID": "50",
8   "PersonType": "Mand",
9   "Note": "blabla",
10  "Email": [
11    {
12      "EmailAdresse": "dannytran_dk@hotmail.com"
13    }
14  ], "Telefon": [
15    {
16      "TelefonNummer": "26 85 15 86",
17      "TelefonType": "Privat",
18      "TeleSelskab": "OISTER"
19    }
20  ], "FolkeRegisterAdresse": {
21    "GadeNavn": "Finlandsgade 22",
22    "By": {
23      "Bynavn": "Aarhus N",
24      "Postnummer": "8200",
25      "Land": "Danmark",
26    }
27    "ZIP": {
28      "Postnummer": "8200",
29      "Land": "Danmark",
30      "Bynavn": "Aarhus N"
31    }
32  }, "AlternativAdresse": [
33    {
34      "GadeNavn": "Finlandsgade 22",
35      "AdresseType": "Arbejde",
36      "By": {
37        "Bynavn": "Aarhus N",
38        "Postnummer": "8200",
39        "Land": "Danmark",
40      }
41      "ZIP": [{
42        "Postnummer": "8200",
43        "Land": "Danmark",
44        "Bynavn": "Aarhus N"
45      }]
46    }
47  ]
48 }
```

Listing 1: JSON-udsnit

## 4.2 XML

```

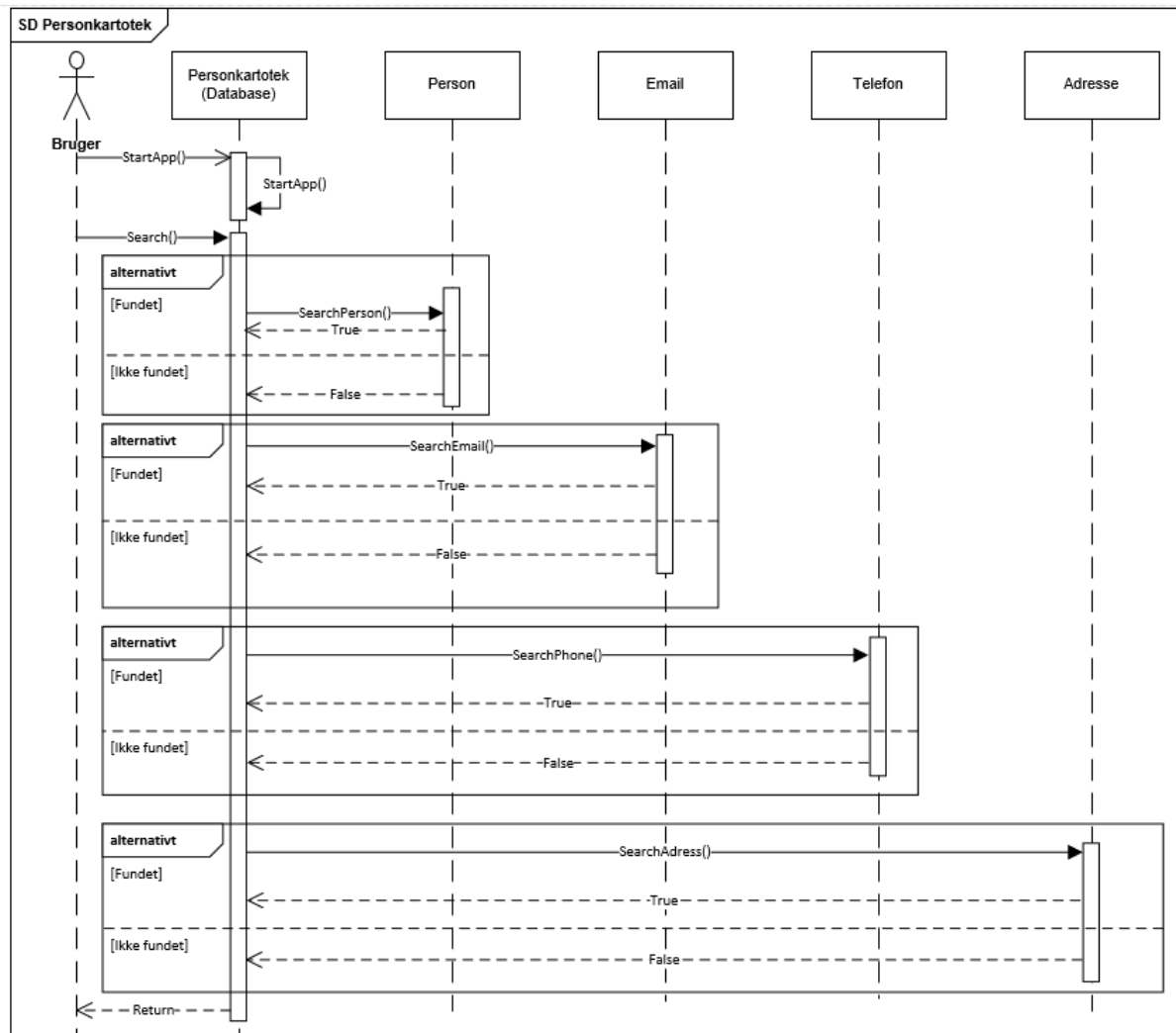
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <Person>
3   <Navn>
4     <ForNavn>Danny<ForNavn/>
5     <MellemNavn>something<MellemNavn/>
6     <EfterNavn>Tran<EfterNavn/>
7   <Navn/>
8   <PersonID>50<PersonID/>
9   <PersonType>Mand<PersonType/>
10  <Note>blabla <Note/>
11  <Email>dannytran_dk@hotmail.com<Email/>
12  <Telefon>
13    <TelefonNummer>26 85 15 86<TelefonNummer/>
14    <TelefonType>Privat<TelefonType/>
15    <TeleSelskab>OiSTER<TeleSelskab/>
16  <Telefon/>
17  <FolkeRegisterAdresse>
18    <GadeNavn>Finlandsgade 22<GadeNavn/>
19    <By>
20      <Bynavn>Aarhus N<Bynavn/>
21      <Postnummer>8200<Postnummer/>
22      <Land>Danmark<Land/>
23    <By/>
24    <ZIP>
25      <Bynavn>Aarhus N<Bynavn/>
26      <Postnummer>8200<Postnummer/>
27      <Land>Danmark<Land/>
28    <ZIP/>
29  <FolkeRegisterAdresse/><AlternativAdresse>
30    <AA>
31      <Type>Arbejde<Type/>
32      <By>Aarhus N<By/>
33      <Postnummer>8200<Postnummer/>
34      <GadeNavn>Finlandsgade 22<GadeNavn/>
35      <ZIP>
36        <By>Aarhus N<By/>
37        <Postnummer>8200<Postnummer/>
38        <Land>Danmark<Land/>
39      <ZIP/>
40    <AA/>
41    <AA>
42      <Type>Hjem<Type/>
43      <By>Aarhus V<By/>
44      <GadeNavn>Trillegaardsvej 85<GadeNavn/>
45      <ZIP>
46        <By>Aarhus V<By/>
47        <Postnummer>8210<Postnummer/>
48        <Land>Danmark<Land/>
49      <ZIP/>
50    <AA/><AlternativAdresse/>
51 <Person/>

```

Listing 2: XML-udsnit

## 5 Overordnet SW-Arkitektur

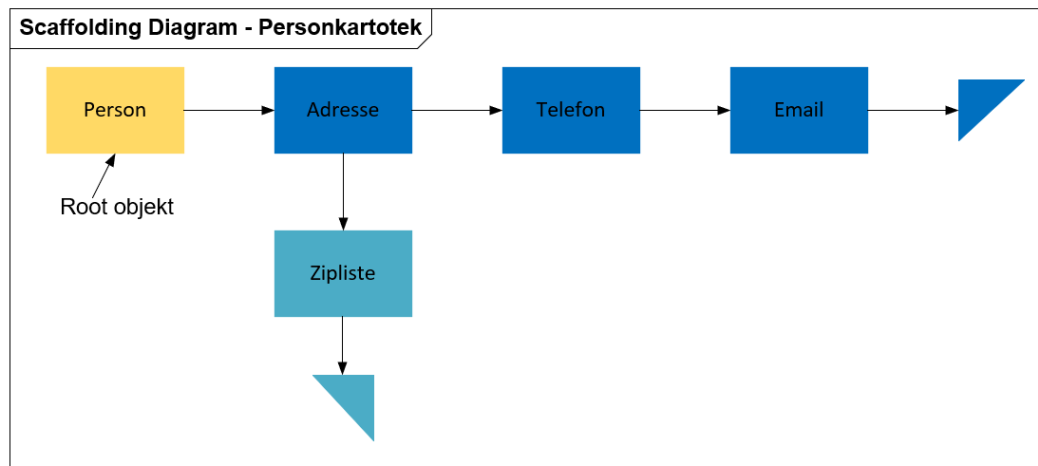
Her ses et sekvens diagram, som giver et overblik over det overordnet SW-arkitektur.



Figur 3: Sekvensdiagram over arkitekturen

## 6 Scaffolding-diagram & Graf

### 6.1 Scaffolding-diagram

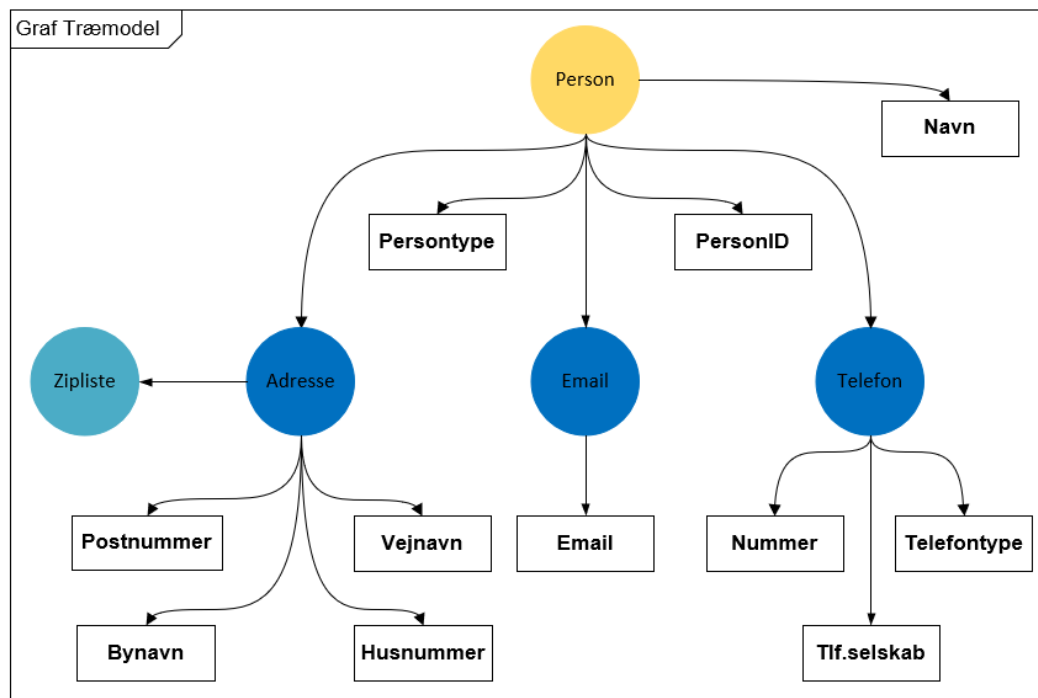


Figur 4: Scaffolding Diagram

Som man kan se på Figur 4 har vi Person som er sat til at være root-objektet. De mørkeblå kasser er objekter som er en samling under Root. Den lyseblå kasse er et objekt på et objekt, som er under Root. Trekanterne repræsenterer afslutningen på en collection/container.



## 6.2 Graph



Figur 5: Graph

Som man kan se på Figur 5 har vi Person som Root igen.