## 4. Iteration 2

I følgende kapitel beskrives use casen ”Handle DVD - CRUD”, og ud fra denne skrives SSD med systemhændelser og tilhørende operationskontrakter. Ud fra use casen og SSD’en udarbejdes interaktionsdagram, hvorefter vores design klasse diagram opdateres.

### 4.2 Use case Handle DVD - CRUD

createDvd ”C”

**Primær aktør:** User

**Pre-betingelse:** ingen

**Post-betingelse:** Et dvd objekt er oprettet

**Hoves succes scenarie:**

* Brugeren angiver DVD oplysninger(id, titel, director, releaseDate)
* Systemet accepterer oplysningerne og opretter en ny dvd.

findDvd ”R”

**Primær aktør:** User

**Pre-betingelse:** Dvd er oprettet

**Post-betingelse:** Dvd er fundet

**Hoves succes scenarie:**

* Brugeren angiver det rigtige id
* System accepterer oplysningerne og finder Dvd.
* System returner det rigtig svar.

updateDVD ”U”

**Primær aktør:** User

**Pre-betingelse:** Dvd er fundet

**Post-betingelse:** Systemet opdater Dvd

**Hoves succes scenarie:**

* Brugeren finder dvd, ændre op oplysninger.
* Systemet accepterer oplysningerne og opdater dvd.

deleteDvd ”D”

**Primær aktør:** User

**Pre-betingelse:** Dvd er fundet i systemet og eventuelle eksemplarerne er ikke udlånt

**Post-betingelse:** Systemet har slettet dvd og eventuelle eksemplarer

**Hoves succes scenarie:**

* Brugeren finder dvd, og sletter dvd.
* System accepterer oplysningerne og sletter dvd og eventuelle eksemplarer

Ovenstående use case er skrevet som CRUD, og giver anledning til følgende SSD for Handle DVD - CRUD.



  
Ovenstående SSD har følgende operationskontrakter.

## Interaktionsdiagrammer

Som det først interaktionsdiagram skal dette nye interaktionsdiagram vise, hvordan de metoder som er beskrevet i SSD kommunikere med de forskellige klasser. Det skal være med til at giv et overblik, for at vi skal løse designklassediagrammet. Dette interaktionsdiagram er større fordi der er to CRUD med i en. Det skyldes at Dvd skal have forskellig Copy’s ligesom at DvdCollection skal havde forskellig Dvd’er. Så det har givet et bedre overblik at samle det i en.

Sekvensdiagram viser næste det samme som i den først bare det er over Dvd i stedet for. Der er de samme metoder CreateDvd, addDvd , getDvd , deleteDvd og updateDvd. Det foregå ligesom med Person objekt. Bare det er over et Dvd Objekt.

Det nye i dette sekvensdiagram er at klasse Copy også er med, det nye i create er, når man holder et objekt af Copy, skal man bare først finde den dvd, som man ville have copy lagt til dvd. Så der bruger vi en metode som er lavt før der hedder getDvd(id), fordi hver dvd har et forskelligt id. Ligesom copy for et forskellig serialNo. Når det dvd objekt er fundet bliver det add en copy til Dvd objekt. Ellers fungere de andre metoder getCopy, deleteCopy og updateCopy på samme måde som de ander bare at man søger i en bestem dvd som man har fra DvdCollection.



Designklassediagrammet.

Ligesom det først Designklassediagram skal vi bruge det fra interaktionsdiagrammet, som har givet et overblik over hvilke klasse, som skal bruge hvilke metoder. Forskellen på dette designklassediagram er at klassen dvd indeholder forskellig copy’s men det er stadig DvdCtr som styre det hele, og har synlighed til det hele. Der er stadig Ekspert mønstret, HighCohesion og lav kobling som bliver brugt fra GRASP. Ellers er der ikke den stor forskelle på de to diagramme ud over navn på klassen, attributter og metoder.

