Analyse af CRUD Loan, hvor der er blevet lavt en fully dressed beskrivelse af CRUD, hvor der er en beskrivelse af primær aktør, pre-betingelse, post-betingelse og et hoves succes scenarie. Det er blevet gjort for at giv et bedre overblik over hvad systemet skal.

## Use case: Håndter Loan – CRUD

createLoan ”C”

**Primær aktør:** Brugeren

**Pre-betingelse:** Personobjekt og Copyobjekt findes i forvejen.

**Post-betingelse:** et loanobjekt er oprettet og associeret med personobjekt og copyobjekt, hvis det ønskede copyobjekt ikke er udlånt i forvejen

**Hoves succes scenarie:**

* En bruger henvender sig for at låne.
* Person navn eller id inddateres.
* System finder personen.
* Det angives hvilket copy der ønskes lånt.
* Systemet registrerer copy og fortæller at loan er oprettet.

getLoan ”R”

**Primær aktør:** Brugeren.

**Pre-betingelse:** et loanobjekt er oprettet.

**Post-betingelse:** et loanobjekt er fundet.

**Hoves succes scenarie:**

* Brugeren søger på et loan.id.
* Systemet returner et loanobjekt på det søgte id.

updateLoan ”U”

**Primær aktør:** Brugeren.

**Pre-betingelse:** et loanobjekt er fundet.

**Post-betingelse:** Systemet opdaterer loanobjektet.

**Hoves succes scenarie:**

* Brugeren finder et loanobjekt og ændrer oplysninger.
* Systemet accepterer oplysningerne og opdaterer loanobjektet.

deleteLoan ”D”

**Primær aktør:** Brugeren.

**Pre-betingelse:** et loanobjekt er fundet i systemet.

**Post-betingelse:** Systemet har slettet loanobjektet.

**Hoves succes scenarie:**

* Brugeren sletter det fundne loanobjekt.
* System accepterer brugerens ønske, og loanobjektet bliver slettet.

## Systemsekvensdiagram og operationskontrakter

## Interaktionsdiagrammer

I følgende interaktionsdiagrammer som et sekvensdiagram, som giver et bedre overblik over hvordan systemet skal kommuniker sekvensdiagram viser, hvordan vores controller kommunikere med modellaget, hvor CRUD bliver vist frem, for at create et Loan er vi nød til først at finde en Person og en Copy, når vi har de to objekter kan vi create et Loan med de nødvendige oplysninger. Read at man søger på det id som Loan har. Update er også at søge på id og så redigere de nye oplysninger og delete er også at søge på id og finde et objekt frem og så fjerne det.



Designklassediagrammet.

I designklassediagrammet giver det et bedre overblik over metoder, datatyper og synligheden, viser også hvordan GRASP bliver brugt. Det som man kan se ud fra diagrammet hvor GRASP bliver synlig gjort er at der lagt vægt på Ekspert mønstret, som tildelt ansvaret til det objekt der har informationen til at fuldføre det. High Cohesion mønstret som betyder lav binding mellem metoderne.

Designklassediagrammet for alle klasser her er angivet de nødvendige attributter og metoder til de forskellige klasser. Vores nye controller har synlighed til DvdCollection, LoanCollection, AddressBook og Loan, for at samle CRUD metoderen i controllerlaget, da det giver et bedre overblik. 