Arbeidskrav 1   
DAT1000

# USN Vår 2022 Gruppe 1

# 

Erik Bøhle - 243905

Didrik Ruud Sawkins - 249950

Robin Tangen - 249952

Vegard Sveinsvoll - 249963

Roman Kollar - 251538

[**Individuell konseptuell datamodell**](#_5l7gyslcpz90) **3**

[Didriks besvarelse](#_4emdff6i9vog) 3

[Vegards besvarelse](#_370hm57dkuzh) 3

[Romans besvarelse](#_xz53jau7nkx1) 4

[Robins besvarelse](#_o3azjikr5zfw) 4

[Eriks besvarelse](#_pkrw77sob0ad) 5

[**Felles konseptuell datamodell**](#_sdme8pwl3o4) **6**

[**Felles entitetisert konseptuell datamodell med atributtliste**](#_ga9om339nm41) **7**

[**Individuelle faglige vurderinger**](#_nl7x7u2yqtvk) **8**

[Didriks faglige vurdering](#_4v865wmw4sh5) 8

[Eriks faglige vurdering](#_o6z0bb2obesy) 10

[Vegards faglige vurdering](#_r74wvorpprfv) 12

[Robins fagelige vurdering](#_oo7mu9tgj5ds) 13

[Romans faglige vurdering](#_dcsj23fpuqfo) 14

[**Litteraturliste:**](#_pc80aiveisbv) **16**

# 

# 

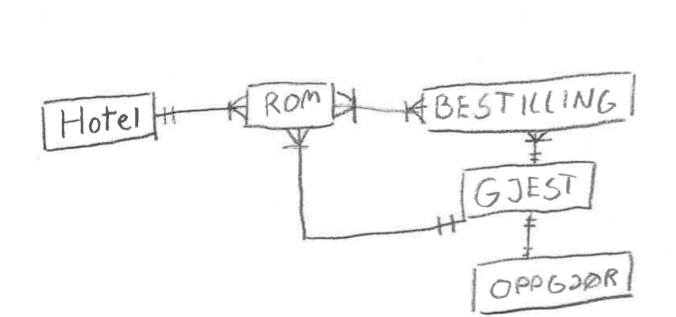
# Individuell konseptuell datamodell

## Didriks besvarelse

# 

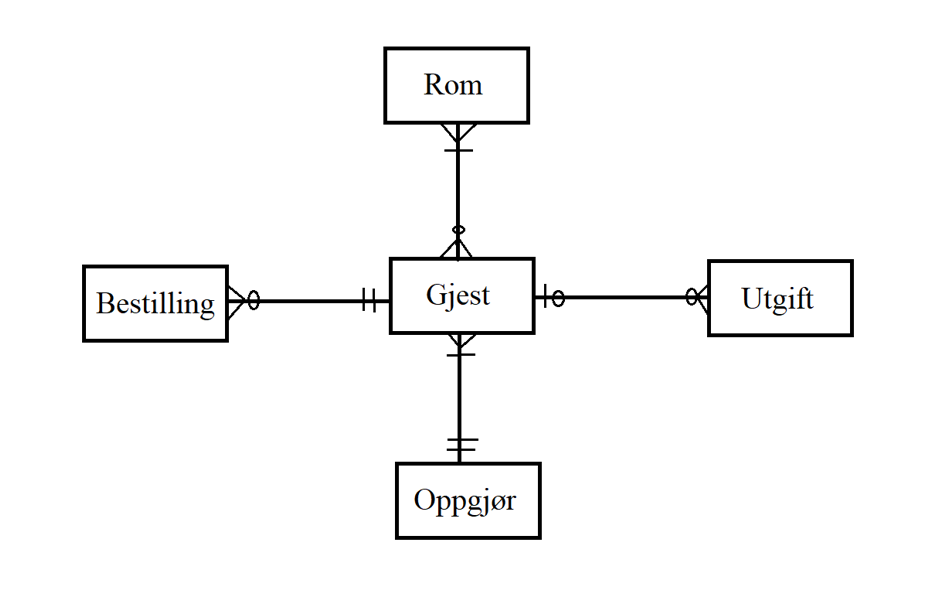
Figur 1

## Vegards besvarelse



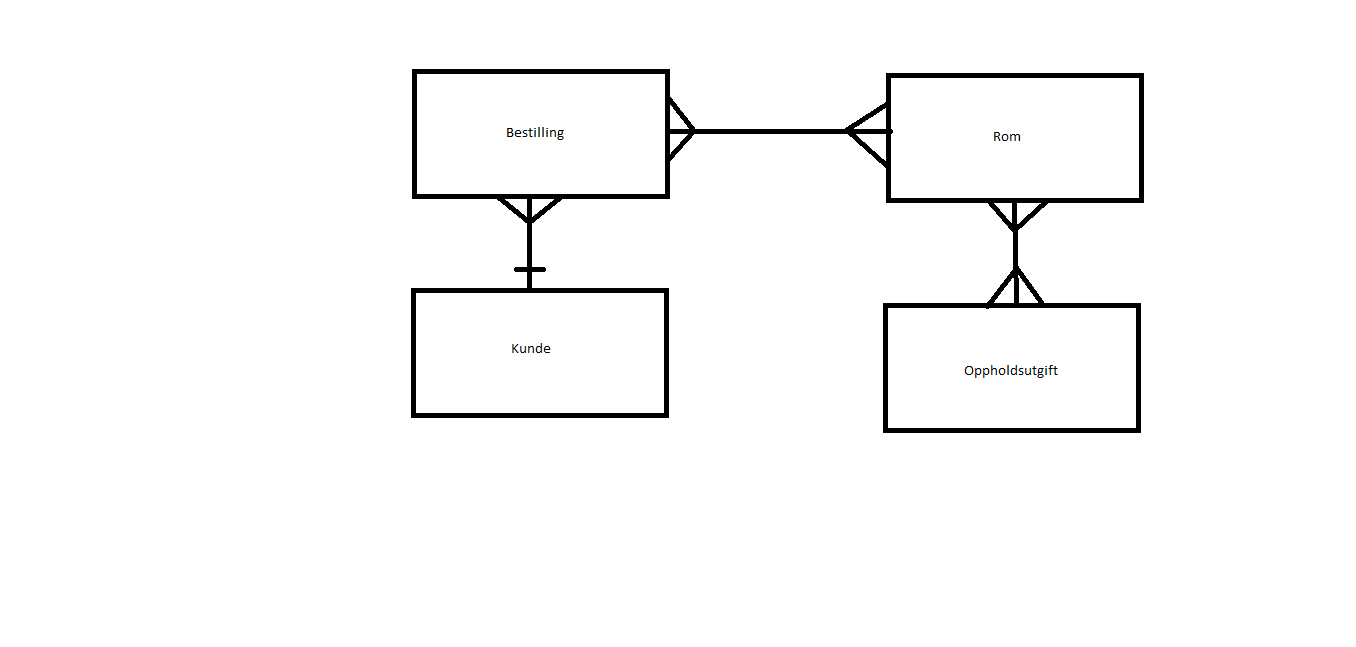
Figur 2.

## Romans besvarelse



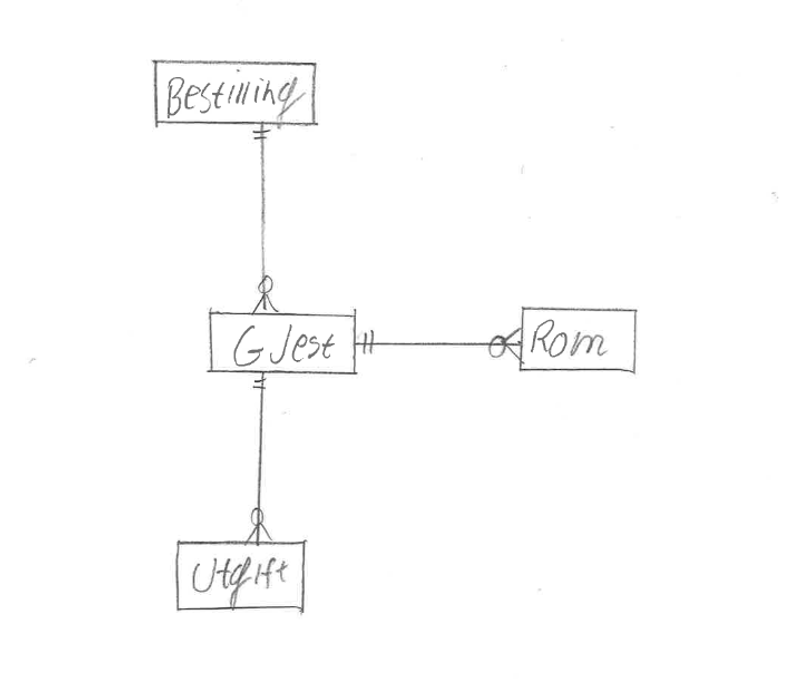
Figur 3.

## Robins besvarelse



Figur 4.

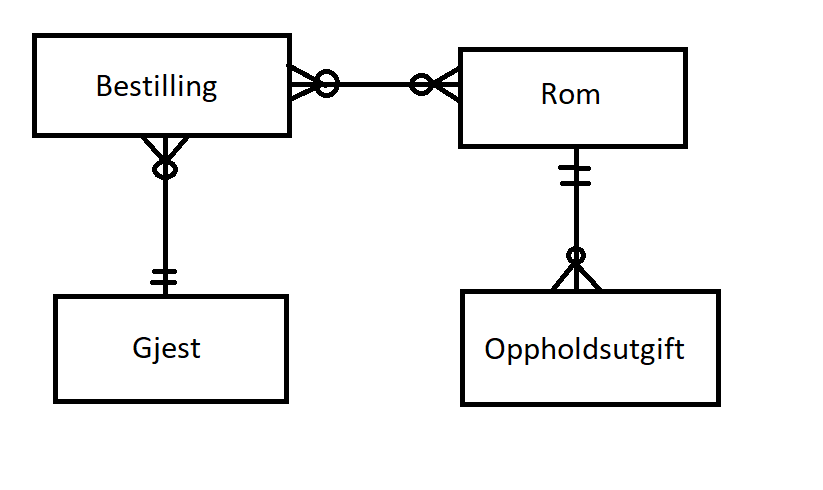
## Eriks besvarelse



Figur 5.

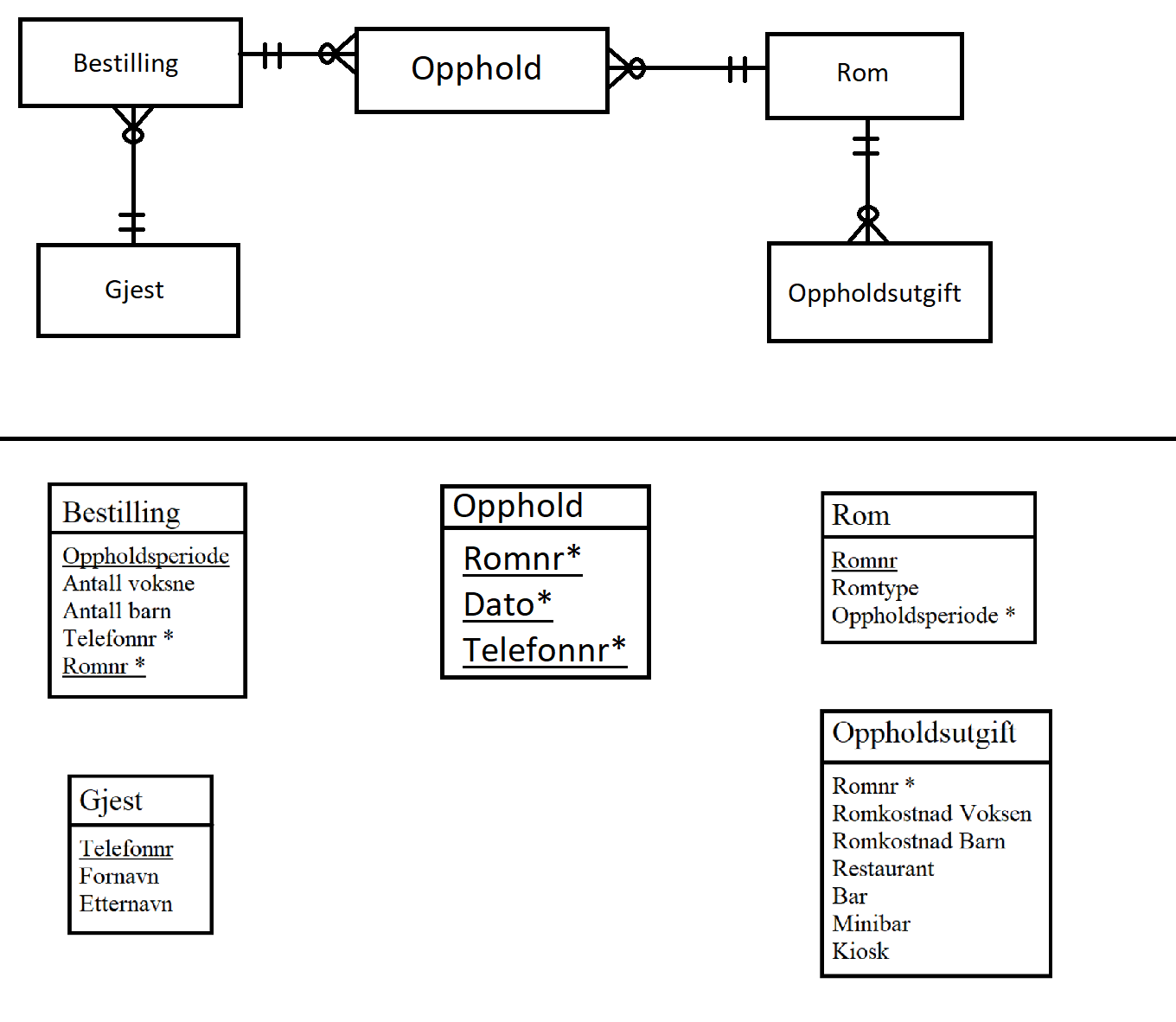
# 

# Felles konseptuell datamodell



Figur 6

# Felles entitetisert konseptuell datamodell med atributtliste

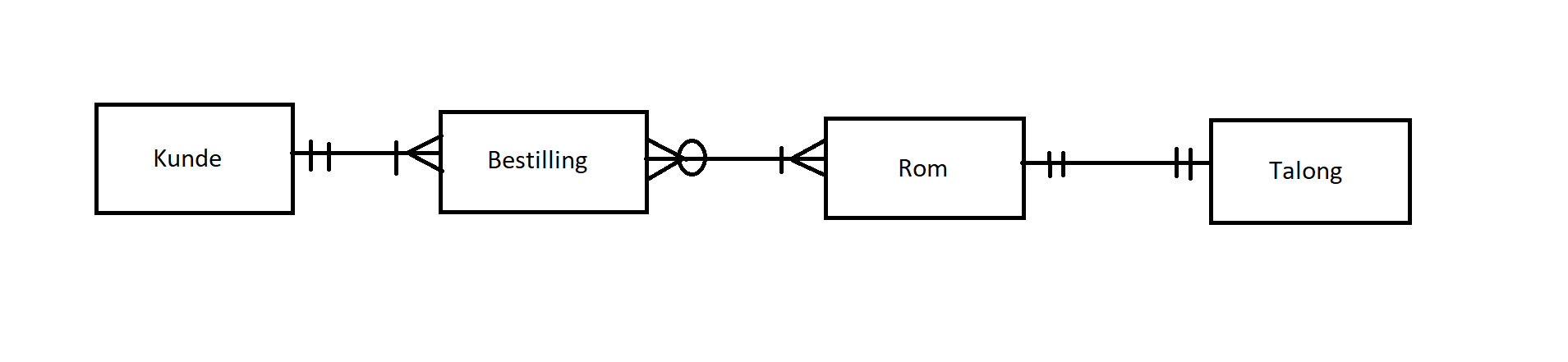


Figur 7.

# 

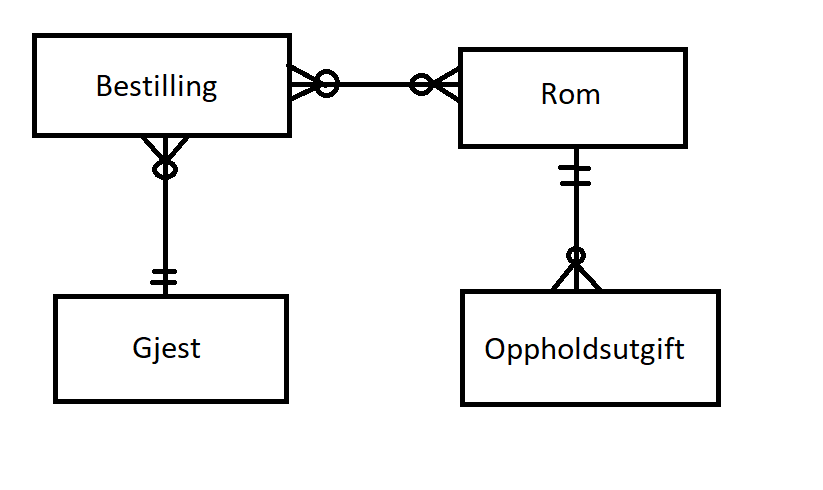
# Individuelle faglige vurderinger

## Didriks faglige vurdering



Figur 8

Det første jeg begynte med var å identifisere entitetene slik de stod beskrevet i oppgaven. Ut ifra teksten identifiserte jeg “kunde”, “bestilling”, “rom” og “talong” som entiteter i databasen. Mellom “kunde” og “bestilling” definerte jeg et 1:n forhold da en kunde vil kunne ha en eller flere bestillinger, men en bestilling vil bare ha en kunde tilknyttet seg. Jeg anså også minimums kardinaliteten på begge til å være minimum 1 på hver side da jeg tenkte det ikke kunne eksistere en bestilling uten en kunde å binde denne opp imot. Mellom “bestilling” og “rom” identifiserte jeg et n:m forhold med minimums kardinal 0 mot bestilling og 1 mot rom. Igjen på bakgrunn av at et rom ikke kunne eksistere uten en bestilling å knytte dette opp imot. Mellom “rom” og “talong” identifiserte jeg et 1:1 forhold på bakgrunn av at et rom kun kunne ha en talong og en talong kunne kun være tilknyttet et rom.



figur 9

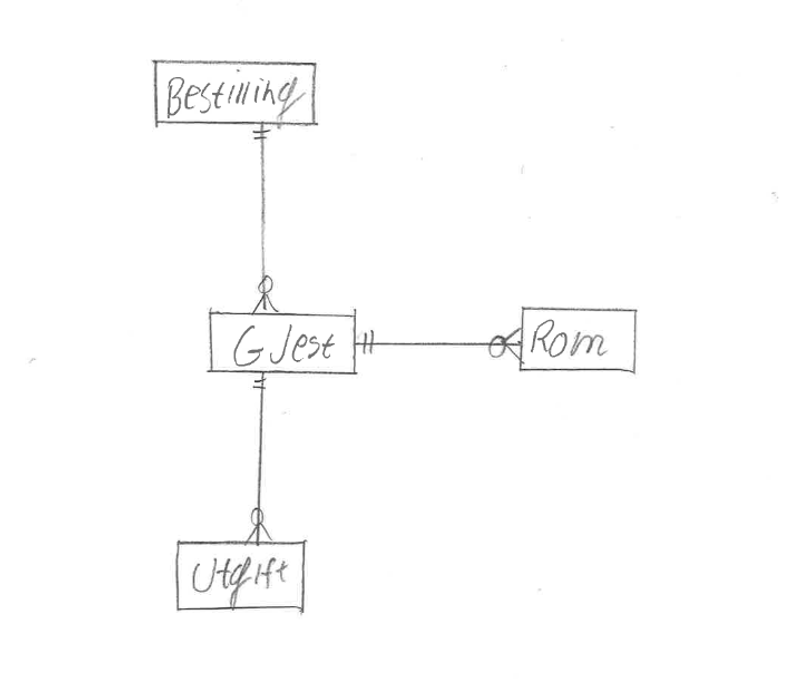
**Sammenligning**

Når vi samlet oss sammen som gruppe og skulle danne en felles konseptuell modell ble det raskt tydelig at jeg hadde noen logiske feil i min første modell. Den første feilen jeg fant var min logikk rundt at “bestilling” ikke kunne eksistere uten “kunde” og at “rom” ikke kunne eksistere uten “bestilling” med tanke på minimums kardinaliteten. En bestilling vil fortsatt kunne eksistere uten kunde tilknyttet bestillingen, samt at et rom vil fortsatt kunne eksistere uten å være bundet opp mot en bestilling. En annen endring vi gjorde var å endre “talong” til “oppholdsutgifter”. Dette ble gjort først og fremst fordi “oppholdsutgifter” er et bedre ord for å beskrive innholdet i listen, men også for å kunne fjerne 1:1 forholdet mellom dette og “rom”. Det ble da mer tydelig at et rom kan ha ingen, en eller flere oppholdsutgifter, så lenge de skilles på tidspunktet for oppholdet, men en oppholdsutgift kan da kun være koblet til ett rom.

**Konklusjon**

Etter dette arbeidet føler jeg at jeg har mer forståelse for datamodellering. Ut ifra min første tegning i figur 1 sammenlignet med vår felles modell i figur 2, føler jeg at jeg var inne på ting med min orginale tankegang, men at jeg nå har forstått hvordan en entitet kan ha ingen, en eller flere forhold med en annen entitet uten at det betyr at den første entiteten ikke kan eksistere.

## Eriks faglige vurdering



Figur 10

I min konseptuelle modell begynte jeg med å identifisere hvilke mulige attributter og entiteter kunne være i casen og lette spesielt etter hvilke entiteter som ikke ga mange til mange forhold. Gjennom dette koblet jeg alle entiteter til gjest, hvor vi har flere gjester som kun har èn bestilling hver, og èn gjest kan kun bo ved et rom per bestilling. Rommet kan dessuten ha flere utgiftsbeløp per gjest, men beløpet skal kun eksistere for den gjesten. Jeg har også modellert minimumskardinalitetene til at en gjest skal kunne registreres i databasen uten at det er en reservasjon, men finnes det en reservasjon skal det finnes en gjest, og om det finnes en gjest behøver det ikke å være en utgift fordi romreservasjonen blir håndtert under eniteten gjest.

I forhold til vår felles konseptuelle modell så har vi valgt å ha et mange til mange forhold mellom rom og bestilling fordi en bestilling kan ha flere rom og vice versa, vi har også valgt å sette en minimumskardinalitet som 0 i alle mange-forhold, fordi hotellet skal ha mulighet til å ikke ha gjester boende. Vi valgte også å sette oppholdsutgifter som en svak entitet koblet til rom, hvor utgifter ikke kan eksistere uten et rom og et rom kan ha forskjellige utgifter. Dette gir også mening om entiteten reservasjon og rom kan inneholde en oppholdsperiode og vi unngår dermed duplikat ved senere entitetisering og input av data.

Jeg mener datamodellene er like, men den datamodellen vi lagde i felleskap gir mer dybde i forhold til informasjon og viser flyten mellom utgifter og opphold klarere.

Ved koblingen mellom rom og gjest gjennom bestilling klarer denne modellen også å vise klarere linjer mellom et opphold og gjesten.

Kort fortalt, den felles modellen gir et klarere bilde av de forskjellige oppholdene, som er noe en oppholdsdatabase bør kunne gjøre.

## Vegards faglige vurdering

**Sammenligning**

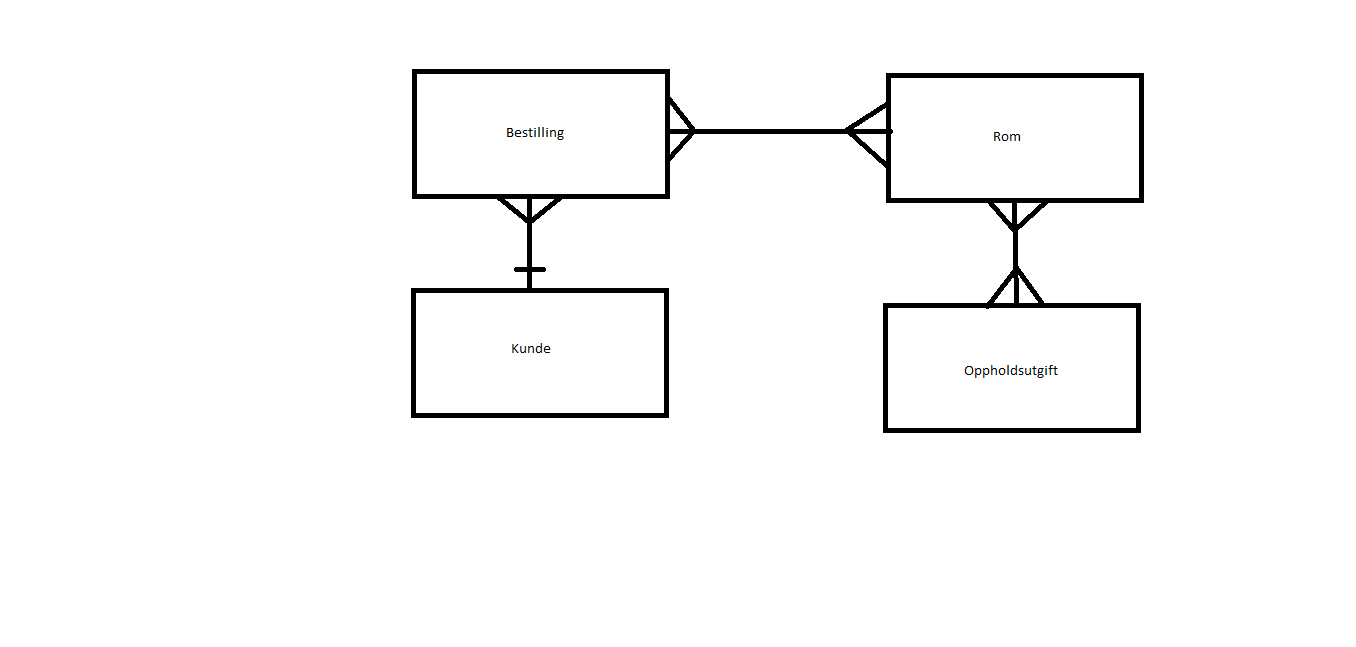
Da jeg lagde min modell lagde jeg den kun basert på ERD symboler. Noe som var manglende i min modell var mangelen på å sette minimumkardinalen på 0 ved flere relasjoner. Når jeg lagde modellen min klarte jeg ikke tenke meg frem til at hotellet, rommene eller gjestene kunne være tomme, som førte til at det alltid ville være 1:m kardinaler i alle relasjonene utenom 1:1 kardinalene. Entitetene “Gjest” og “ Oppgjør” i figur VS ble byttet ut med “Rom” og “Oppholdsutgift” i modellen Gruppe. Modellen gruppe bruker heller “Gjest” i relasjon til “Bestilling”, hvor vi heller baserer oss på at entiteten “Bestilling” blir hoved entiteten som holder mest informasjon, istedenfor “Gjest” i min modell som skulle være hoved entiteten.

Modellen min tar også i bruk entiteten med navn “Hotell”, noe som også ikke blir tatt i bruk i gruppemodellen med tanke på at hotellet fungerer som en aktør og ikke en entitet i modellen.

**Konklusjon**

Diagrammet mitt hadde flere avvik og mangler både i entitet bruk, relasjoner mellom entitetene og kardinalene brukt i relasjonene. Jeg har lært mye om hvordan ting blir satt opp på en logisk måte og bakgrunnen til hvert valg jeg tar, som riktig bruk av kardinaler basert på relasjonene mellom entiteter, entitisering og normalisering når jeg skal bygge et diagram.

## Robins fagelige vurdering



figur 11

Da vi fikk Casen Storenuten, hadde jeg allerede fra før godt kjennskap til hvordan hotellet fungerte, så jeg fikk raskt en ide om hvordan databasen deres skulle være knyttet sammen ut ifra min forståelse av datamodellering.

Jeg tenkte først at vi trengte entiteten bestilling, denne koblet jeg til entiteten rom med n:m relasjon, for så og koble bestilling til entiteten kunde med en 1:n relasjon. Deretter trengte jeg en entitet for å ta alt med utgifter. Denne entiteten kalte jeg oppholdsutgift og knyttet sammen med entiteten rom med n:m relasjon.

Forskjellen fra min modell figur 11 og gruppa sin figur 6 er at jeg hadde entiteten kunde istedetfor gjest. Entitets relasjonen min mellom entiteten rom og entiteten oppholdsutgift var også n:m istedenfor 1:n som vi kom frem til som gruppe at det skulle være. Jeg brukte kunde istedetfor gjest da jeg mener det har ingen betydning hva de blir kaldt i databasen så lenge de blir lagret på rett sted. n:m relasjonen mellom entiteten rom og entiteten oppholdsutgift hadde jeg fordi jeg misforsto sammenhengen mellom dem.

Felles modellen figur 6 er da dratt mest ut ifra min modell figur 11. Selv om vi alle har en tilnærmet lik modell hadde vi alle litt forskjellig utforming på modellen.

## Romans faglige vurdering

Denne oppgaven handlet om å lage en konseptuell datamodell til hotellets innsjekk-, opphold- og utsjekkingssystem.

Jeg begynte oppgaven med å skrive ned stikkord som jeg antok var passende entiteter ut ifra oppgaveteksten. De ordene jeg endte opp med var gjest, bestilling, rom, utgift og oppgjør. Jeg begynte så å tegne en konseptuell datamodell der jeg valgte å ha entiteten «Gjest» i sentrum. Dette på bakgrunn av at det er gjesten vi ønsker å lagre informasjon om i databasen. Deretter koblet jeg de resterende entitetene direkte til gjest. Tanken bak dette var at ingen av de resterende entitetene ville «eksistert» uten at det finnes en gjest. For eksempel vil ikke en bestilling eksistere dersom det ikke finnes en gjest.

Jeg forstod relativt raskt at jeg tenkte litt i feil ende etter at jeg og gruppa mi møttes for å lage en felles konseptuell datamodell. Et eksempel på hvorfor jeg tenkte feil er at entiteten «rom» ville eksistert selv om det ikke finnes en entitet «gjest». I tillegg så jeg at det å ha to forskjellige entiteter for «oppgjør» og «utgift» også var unødvendig, da alt dette kunne samles i en entitet ved navn «oppholdsutgift». Resten av entitetene samsvarte med hva gruppemedlemmene hadde lagd individuelt.

Når det kommer til relasjoner mellom entitetene i min kontra gruppas datamodell, så er det en del forskjeller. Grunnen til at jeg hadde «gjest» i midten av modellen er at jeg tenkte gjest var en slags hoved-entitet.

Relasjonen mellom gjest og bestilling ble at en gjest kunne ha ingen, en eller flere bestillinger, men en bestilling tilhørte en og kun en gjest. Denne relasjonen mellom entitetene stemmer med vår felles datamodell.

Relasjonen mellom gjest og rom ble at en gjest kunne bestille ingen, en eller flere rom, og et rom kunne ha ingen, en eller flere gjester. Allerede her kunne jeg se at entiteten «bestilling» burde vært en relasjon mellom entitetene gjest og rom, slik vi gjorde i vår felles datamodell.

Videre var relasjonen i min datamodell mellom gjest og utgift, der utgiften var alle ekstra utgifter fra restaurant, bar etc. i løpet av Oppholdet. Relasjonen kunne beskrives slik at gjesten kunne ha ingen, en eller flere utgifter mens en utgift kan ha ingen eller maks en gjest. Grunnen til at den kunne ha maks en gjest var at utgift på daværende tidspunkt kun var knyttet til den konkrete gjesten.

Min siste relasjon i den konseptuelle datamodellen var mellom gjest og oppgjør der gjesten måtte ha et og kun ett oppgjør, mens oppgjøret måtte tilhøre i hvert fall en eller flere gjester. Som nevnt tidligere burde entitetene utgift og oppgjør slås sammen til entiteten oppholdsutgift.

# Litteraturliste:

Kristoffersen, B.(2020). *Databasesystemer* (5. utg.)Oslo: Universitetsforlaget