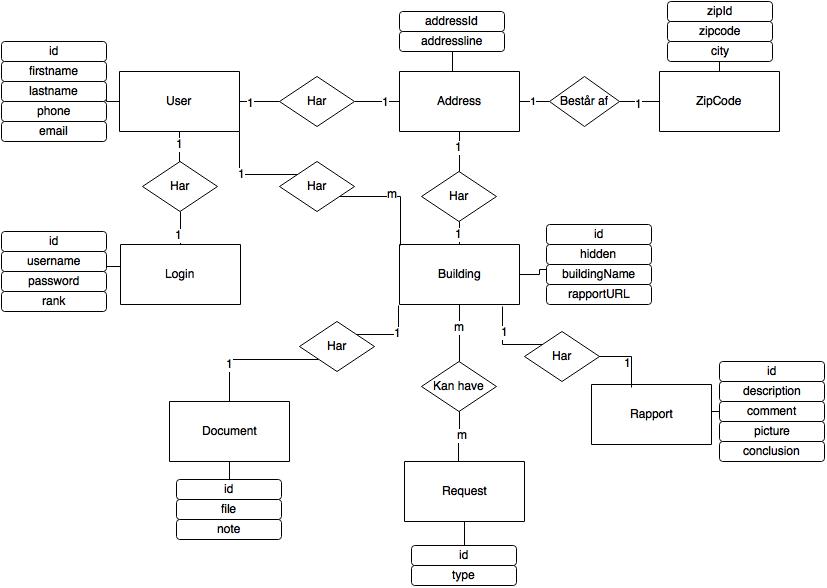
# ER Diagram

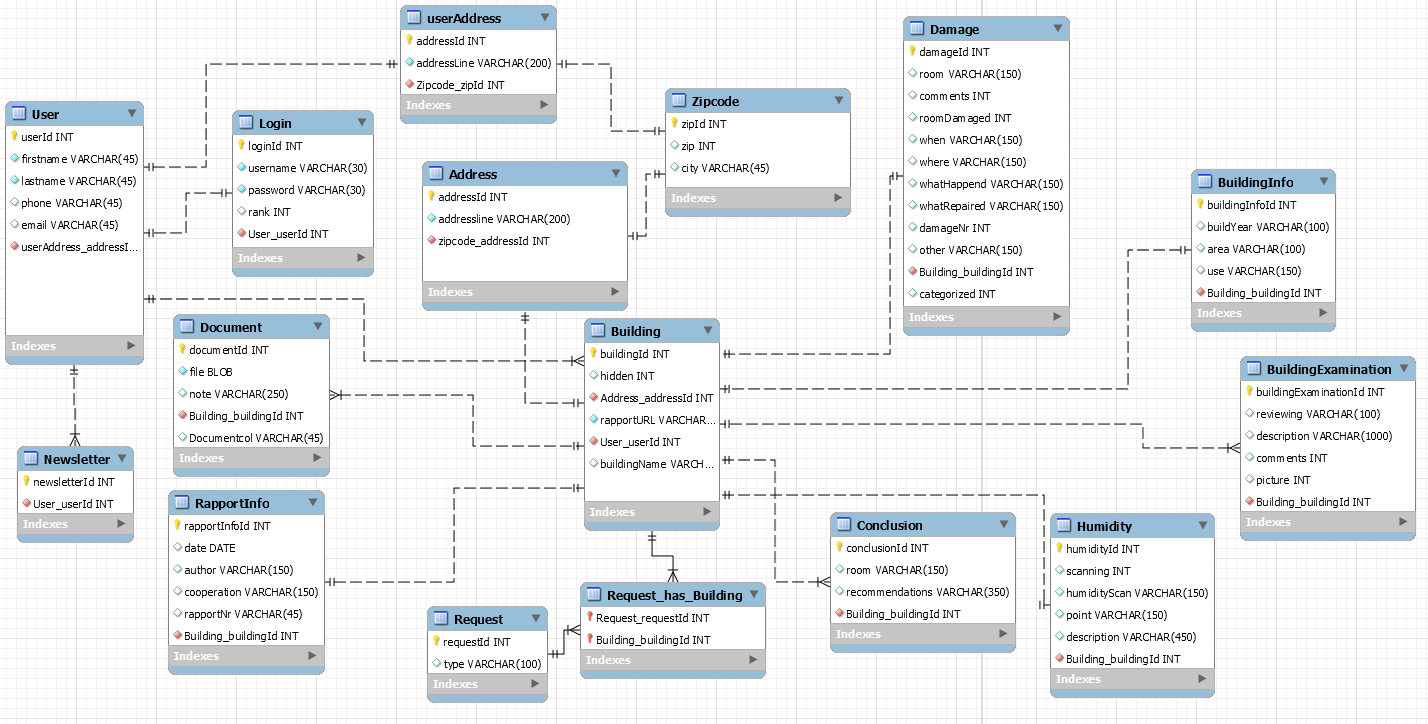
Vi bruger ER diagrammet til at illustrere de forskellige relationer mellem tabellerne, som skal oprettes i databasen. Det har givet os en overordnet ide om hvordan relationerne skal gå og hvilken type de skal være.



# Database model

Ved hjælp fra vores ER diagram, har vi udarbejdet en database model. Der er dog én stor ændring; vi har valgt at dele rapport tabellen op i 6 tabeller, da meget af den store data ikke ville give mening at have i samme tabel. Derudover havde vi behov for at splitte noget af dataen ud over flere tabeller som fx conclusion tabellen, da der kan være flere konklusioner på rapporten.  
Da vi har brugt Polygons sundhedstjek rapport design, er det kun muligt at gå i dybden med et rum pr. bygning, så der er ikke behov for en tabel til de forskellige rum.  
Det er dog muligt at man kan nøjes med færre tabeller til rapporten. Til næste gang ville jeg nok sætte RapportInfo tabellen som bindeled mellem Buildings og alle rapport tabellerne, så alle relationerne går til RapportInfo og RapportInfo har en 1 til 1 relation med Buildings.

SQL scriptet der opretter databasen og alle tabellerne er vedhæftet som bilag.



# SQL

Vi har mest gjort brug af enkle sql queries, da der ikke har været behov for andet.  
En query vi har brugt meget er SELECT Buildings med JOIN på både Address og Zipcode tabellerne. Denne query henter en række data ud fra Building, Address og Zipcode tabellerne, via JOIN.  
JOIN bliver brugt til at sammensætte rækker fra flere tabeller, baseret på ens værdiger i begge tabeller.

For at tage nedenstående query, så henter vi en række data ud fra Buildings hvor buildingId er et givet parameter, med en JOIN på Address hvor Building.Address\_addressId er det samme som Address.addressId og en JOIN på Zipcode hvor Address.zipcode\_addressId er det samme som Zipcode.zipId.



Dataen vi vil få ud fra databasen med overstående query, hvis buildingId er 1.



Asger Lundblad