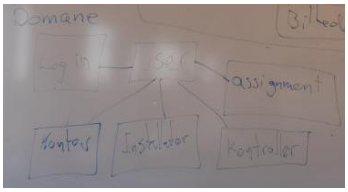
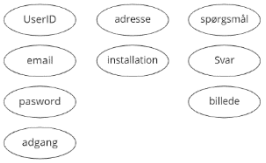
E/R Diagram (Kristian Kristensen)

E/R analyse





For at designe databasen har vi analyseret behovet for attributter ud fra domænemodellen. Vores første domænemodel fokuserede på bruger, med mange forskellige slags brugere. Men vi fandt hurtigt ud af at alle typerne kan gå under én slags bruger med forskellige rettigheder.

Desuden kom der krav til tjeklisterne om blandt andet at indeholde billeder. Derfor har vi fået attributterne: brugerID email pasword til bruger adgang, og adgang som adgangskontrol. På tjeklisten som i vi i starten kaldte for assignment er der flere attributter, men nogle vigtige er adresse og installation, spørgsmål og svar og billeder.

E/R diagram

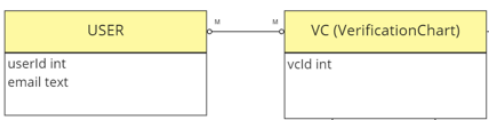
Efter at vi har fået oversigt over attributterne, også kaldet egenskaber, kan vi udarbejde et E/R diagram. Diagrammet giver en god oversigt over relationerne mellem de forskellige enheder i databasen. Vi har valgt at benytte diagrammet for at få placeret de forskellige enheder på en god måde i forhold til hinanden.

Egentligt er programmet ikke afhængigt af at benytte en database, da det er muligt at udfylde tjekskemaet og med det samme at omdanne det til en pdf og at sende det på mail. Men fordelen ved databasen er at den kan langtidsopbevare dataene, som ellers vil gå tabt når programmet lukkes. Ved at langtidsopbevare dataene bliver det muligt for brugeren at stoppe udfyldningen af skemaet, lukke programmet, og at åbne programmet igen på et andet tidspunkt og fortsætte med at udfylde skemaet.

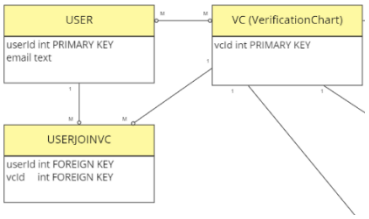
Databasens formål centrerer sig om at kunne gemme indtastningerne til tjekskemaet. Der er dog en fordel mere ved at benytte databasen, vi kan nemlig centralisere dataene. Det betyder at vi i stedet for at gemme indtastningerne fra tjekskemaerne lokalt på den enhed som indtastningen er sket på, kan have dataene gemt i skyen, og lade brugeren tilgå dataene i skyen via sin enhed. Formålet med denne placering af databasen er at der nu kan være flere brugere som har adgang. Det betyder også at der ikke vil være brug for afsendelse af e-mails mellem medarbejdere, da alle medarbejdere vil kunne se tjekskemaerne.

Den centrale placering af databasen giver dog en sikkerhedsudfordring og må derfor beskyttes med adgangskode. Håndteringen af adgangskoder og brugere skal databasen også forberedes til.

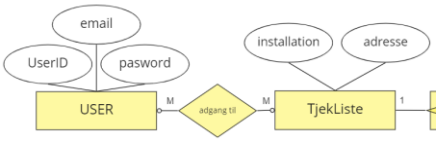
Nedenfor ser du to vigtige entiteter, eller enheder i databasen, tabellen for brugere og tabellen for tjekskemaerne. Vi har gjort os overvejelser om hvor mange brugere der er pr tjekskema, og hvor mange tjekskemaer der er pr bruger. Konklusionen var at vi gerne vil gøre det muligt for brugeren at have flere tjekskemaer at arbejde på samtidig, uden at ét tjekskema skal færdigmeldes før et nyt kan åbnes. Derfor er tabellen for tjekskemaer (VC) markeret med et M for mange. En bruger kan altså have adgang til mange tjekskemaer.

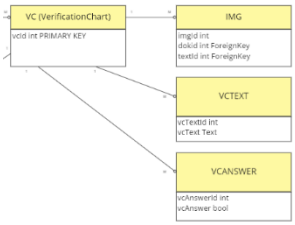


Bruger tabellen (USER) er også markeret med M for mange. Det betyder at flere brugere kan have adgang til det samme tjekskema. Det betyder ikke at alle brugere får adgang til alle tjekskemaer, men databasen rummer muligheden for det. Derfor vil det næste være at lave en begrænsning på hvem som har adgang til hvilke tjekskemaer. Fordi brugerne har adgang til et udefineret antal tjekskemaer, og hvert tjekskema kan have et udefineret antal brugere, skal tabellen for adgangsbegrænsning tage højde for det og må nødvendigvis være variabel. Den lægges ikke sammen med hverken bruger eller tjekliste, men oprettes som en selvstændig tabel som sammenkæder bruger med tjekliste. Bruger tabellen (USER) er oprettet med et id som refererer til den specifikke bruger, dette id markeres med Primær nøgle (PRIMARY KEY) fordi tabellen kan stå alene, det samme gælder for id'et for tjeklister. I join tabellen som sammenkæder brugere med tjeklister sættes to værdier, id'et som refererer til en bruger, og id'et som refererer til en tjekliste. Fordi begge id først er defineret i andre tabeller, markeres de i join tabellen som fremmed nøgler (FOREIGNKEY). Det medfører at join tabellen nu er afhængig af de andre to tabeller, den har ikke nogen funktion hvis den



stå alene. Det betyder at den kun bruges hvis der er oprettet en bruger den kan få et bruger id fra, og en tjekliste som den kan få et tjekliste id fra. Slettes en bruger eller en tjekliste som join får en fremmed nøgle fra, skal rækken i join tabellen altså også slettes. Man kunne måske ønske at gemme historik over hvilke brugere som har haft adgang til hvilke tjekskemaer på hvilke tidspunkter, men det rummer join tabellen ikke information om når en bruger eller tjekliste er slettet. Vil man gemme en historik, bliver det i det her tilfælde i en helt anden tabel, det har vi dog ikke planer om i øjeblikket. Eftersom at tabellen UserJoinVc er relationen mellem de to tabeller tegnes den som en rombe der står på en spids. Den er nu selve relationen mellem de to og ikke en selvstændig tabel. Det fremgår af det nye diagram som har fået lidt andre navne for nemheds skyld.

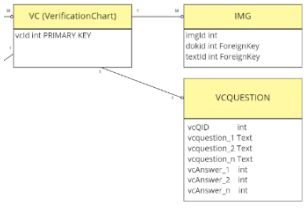




Vi har overvejet om teksterne og svarene til tjeklisterne skulle have en kolonne per spørgsmål eller om det skulle være en række pr spørgsmål som så skulle kæde spørgsmål og svar sammen. Billedet til højre, er i den første del af planlægningen.

Men eftersom hvert svar har en direkte relation til hvert spørgsmål, var det naturligt at lade dem relatere til hinanden. Der kan måske være et spørgsmål uden et svar, men det giver ikke mening at have et svar uden et spørgsmål. Derfor fik spørgsmåls tabellen i starten lov at beholde sin relation til tjekliste, mens svar tabellen relaterede til spørgsmåls tabellen.

Spørgsmålene er foruddefinerede og for hvert spørgsmål er der et svar, svaret kan være ikke udfyldt, ja nej eller irrelevant, men der er kun et svar. Derfor ændres måden svaret lagres på til tal, så der kan lagres 0 1 2 eller 3 for ikke udfyldt, ja nej eller irrelevant. Relationen blev en én til én relation, det betød at der ikke skulle være to tabeller men kun en tabel, som indeholder alle data for de to tidligere tabeller.

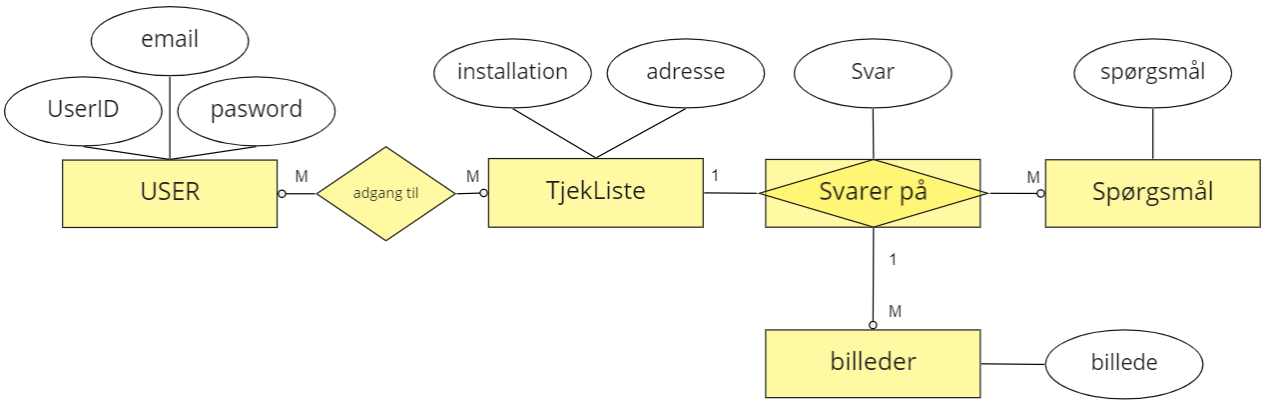


Spørgsmålstabellen kom til at indeholde alle spørgsmål og alle svar. Den reserverede altså meget plads i databasen. Der er bare et problem, for spørgsmålene er foruddefinerede, og behøver ikke at blive gentaget. Man kunne argumentere for at der skal være rum for at gemme historikken med gamle tekster hvis teksterne bliver ændret, men det sørges der for når der afslutningsvist gemmes i pdf, og vi undgår redundans i databasen.

Der er undervejs kommet endnu en oplysning fra projektstiller, nemlig at der skal kunne tilføjes spørgsmål. Det kræver at databasen bliver mere dynamisk og tillader ikke at lægge svarene ind under tjeklistetabellen som ellers var oplagt når det så ud til at tjekliste og svar ville få én til én relation.

Designet af spørgsmål svar og Billeder er nu blevet revurderet på baggrund af behovet for den mere generiske tilgang.

Det har været svært ikke at lave et kompromis når spørgsmål svar og billeder har skullet tilføjes databasen.



Spørgsmålene er tænkt som generiske sådan at der kan tilføjes og slettes spørgsmål efter behov. Spørgsmålene kan genbruge på flere tjeklister, og derfor er relationen mellem tjekliste og spørgsmål en mange til mange relation. Du har måske allerede bemærket at det ikke er det som står i diagrammet, her er den sat som en én til mange relation, og det er der en grund til.

Det er et kompromis, og det er som om at uanset hvordan man vender og drejer det så resulterer det i et kompromis. Der er altså også andre muligheder man kan benytte. Udfordringen ligger i at selvom spørgsmålene kan bruges i mange tjeklister, så vil svaret kun høre til i én tjekliste, og det samme gælder billederne, med endnu en undtagelse, nemlig at de kun hører til i ét bestemt svar i én bestemt tjekliste. Valget er faldet på en udvidet relations tabel. Udover at relations tabellen kobler spørgsmål og tjeklister sammen, så indeholder tabellen også information om svar på spørgsmålet. Der er kun et svar per spørgsmål per tjekliste og nu er svar relationen blevet en én til mange, selvom den som relations kobling egentligt var en mange til mange. Den holdes unik sådan at spørgsmål ikke gentages på samme liste. Man kunne overveje om den i virkeligheden er en selvstændig tabel med en primærnøgle, her er problemet bare at et svar ikke kan eksistere uden et spørgsmål, den er altså afhængig af spørgsmålstabelen. En måde den kunne blive gjort selvstændig på vil være at lade den have sine egne spørgsmål som blev kopieret fra de prædefinerede spørgsmål. Men igen vil det være et kompromis at lave gentagelser af spørgsmålsteksten.

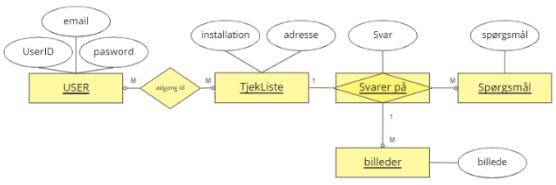
Valget på at lave en kombineret relations tabel med ekstra tabeller er valgt for at gøre databasen så enkel som mulig. At svarene ikke ligger for sig selv ved siden af relations tabellen er fordi det ville være en én til én relation, hvilket vi normalt logger sammen som én tabel.

Det er muligt at lade billederne have relation til både tjekliste og spørgsmål med to fremmednøgler, men for enkelthedens skyld er de her relateret til svartabellen, det gør vi ved at give svartabellen et unikt id som billeder bruger som fremmednøgle. Nu kan vi tilføje et ubegrænset antal billeder til hvert spørgsmål, og vi kan holde styr på hvor hvert billede hører til, så de ikke bliver blandet sammen med billederne fra andre spørgsmål. I praksis vil vi i databasen blot notere stien og navnet på billedet som så gemmes separat i en mappe.

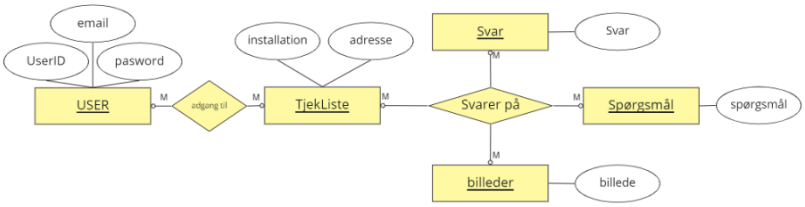
Normalisering

Generelt er der anvendt normalform 1 ved at tabellerne har nøglefelter. Vi har også gjort meget ud af at leve op til normalisering 1 ved at undgå gentagelser. For eksempel var det meget oplagt at have gentagelser af spørgsmålene for hvert tjekskema. Det har vi undgået ved at lave en tabel med spørgsmål og en anden tabel som linker tjekskemaerne med spørgsmålene efter behov. Havde vi lagt de 39 spørgsmål som sikkerhedsstyrelsen har defineret ind i tabellen for tjekskemaet, så ville vi ud over at det ville være voldsomt uoverskueligt også have mange spørgsmål som ikke er relevante eller som aldrig bliver besvaret. Samtidig ville det ikke være muligt at tilføje spørgsmål. Nu kan tjekskemaet variere i antal af spørgsmål uden at hver post varierer.

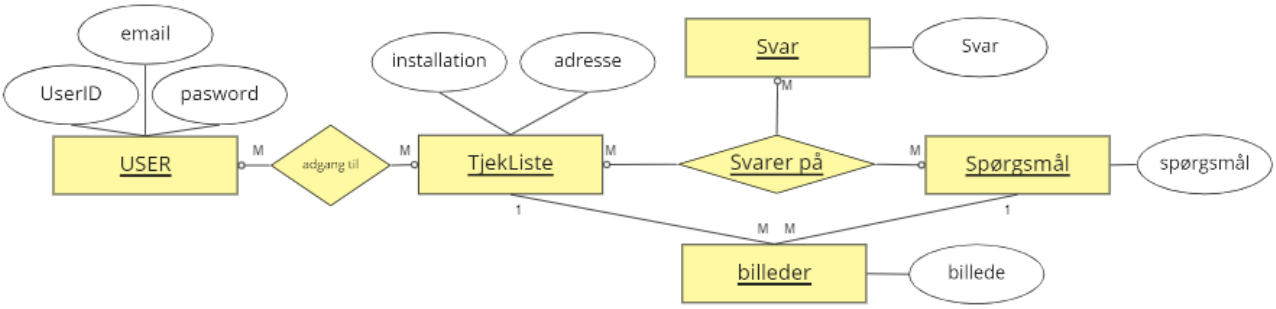
De fleste tabeller opfylder normalform 2 ved at have unikke nøgler. User har et unikt id for hver bruger og Tjekliste har et unikt id for hver tjekliste som genereres. Det samme gælder hvert spørgsmål som oprettes. Lidt anderledes er det for svar tabellen som fungerer som relation mellem tjekliste og spørgsmål og derfor har to fremmed nøgler, de to fremmednøgler kan tilsammen godt udgøre et unikt id, men der kan være en risiko for gentagelser. Gentagelser kan selvfølgelig undgås, men ved at lave et unikt id ved siden af fremmednøglerne, og for nemmere at kunne referere billeder til tabellen vil vi faktisk oprette den med et unikt id så også den lever op til normalform 2. For tabellen for billeder er det vigtigt at den har en fremmednøgle som refererer til et bestemt svar i en bestemt tjekliste, den kan klare sig uden et unikt id, men at tilføje et unikt id vil kun være en fordel da man dermed får mulighed for at lægge billederne i den rækkefølge man ønsker. Havde den ikke et unikt id men kun en gentaget fremmednøgle, ville man stadig kunne finde billederne frem, men rækkefølgen ville være tilfældig. På diagrammet er entiteterne som har primærnøgler nu understregning.



For at leve op til normalform 3 vil vi gerne overveje om der er tabeller som kan deles op i mindre tabeller. Fokus rettes mod relationen svarer på som også er en enhed med svar, disse svar er gentagelser af blandt andet ja og nej, og de lever heller ikke helt op til normalform 1 om at undgå redundans. Vi flytter svar mulighederne ud som en selvstændig enhed, og lader svare på være en ren relation, til fire forskellige enheder.



Vi kan nu svare både ja og nej på et spørgsmål, så vi har inkonsistens. Det løses ved at give relations tabellen en primærnøgle som består at tjekliste spørgsmål og svar, så der kun kan laves en kombination af hver. Nu kan vi kun tilføje et billede til hvert spørgsmål, derfor er vi nødt til at tage billeder ud af relationen Svarer på, og lade billeder selv lave en reference. Vi vælger at lade billeder referere direkte til enhederne tjekliste og spørgsmål. Nu kan vi tilføje adskillige billeder til et bestemt spørgsmål i en bestemt tjekliste uanset om der er svaret på spørgsmålene. relationen til hver enhed er en én til mange og fungerer ved at billeder har en fremmednøgle til hver.



Endelig lever vi også op til normalform 3.

Vi mener også at vi nu lever op til Boyce-code normalform om at hver entitet har attributter som identificerer de andre attributter inden for samme entitet.