

E/R Diagram (Kristian Kristensen)

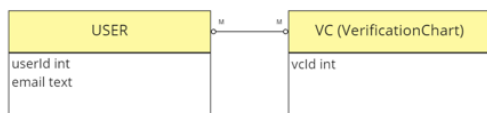
For at designe databasen har vi udarbejdet et E/R diagram. Diagrammet giver en god oversigt over relationerne mellem de forskellige enheder i databasen. Vi har valgt at benytte diagrammet for at få placeret de forskellige enheder på en god måde i forhold til hinanden.

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 fordelene 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.

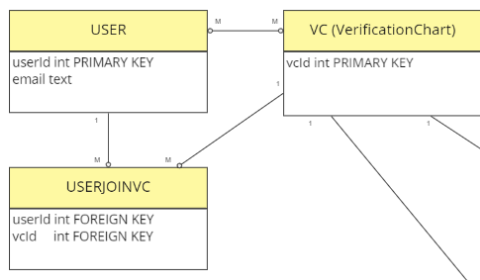
Databasen formål centrerer sig om at kunne gemme indtastningerne til tjekskemaet. Der er dog en fordel mere ved at benytte databasen, vi kan nemlig centralisere dataene. Det betyder at vi i stedet for at gemme indtastningerne fra tjekskemaerne lokalt på den enhed som indtastningen er sket på, kan have 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 tabeller 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, og her kommer den interessante detalje, at når andre brugere har direkte adgang til tjekskemaerne, så bliver det overflødigt at sende tjekskemaer på mail internt mellem medarbejderne. 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

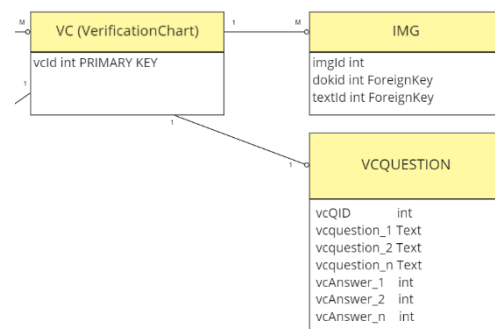
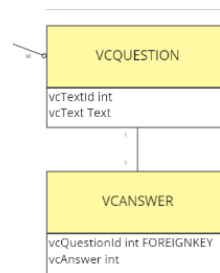
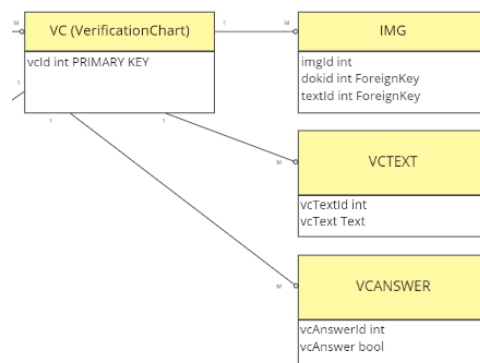


specifikke bruger, derfor markeres dette id med Primær nøgle (PRIMARY KEY), det samme gælder for id'et for tjeklister. I join tabellen som sammenkæder brugere med tjeklister sættes to værdier, id'et som refererer til en bruger, og id'et som refererer til en tjekliste. Fordi begge id først er defineret 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 står 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. Skal man gemme en historik bliver det i det her tilfælde i en helt anden tabel, det har vi dog ikke planer om i øjeblikket.

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 her, er i den første del af planlægningen.

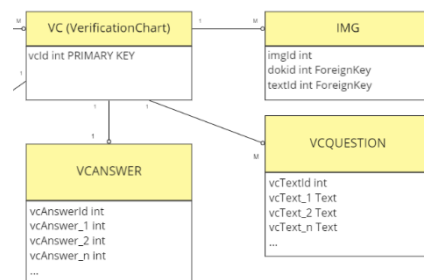
Men eftersom hvert svar (VCANSWER) har en direkte relation til hvert spørgsmål (VCTEXT) vil det være naturligt at lade dem relatere til hinanden. Der kan måske være et spørgsmål uden et svar, men det giver ikke mening at have et svar uden et spørgsmål. Derfor får spørgsmåls tabellen lov at beholde sin relation til tjekliste, mens svar tabellen skal relatere til spørgsmåls tabellen.

Navnet på spørgsmålstabellen er ændret til VCQUESTION som er mere beskrivende. Spørgsmålene er foruddefinerede og for hvert spørgsmål er der et svar, svaret kan være ja nej eller irrelevant, men der er kun et svar. Derfor ændres måden svaret lagres på til tal, så der kan lagres 0 1 eller 2 for ja nej eller irrelevant. Relationen er nu en én til én relation, det betyder at der ikke skal være to tabeller men kun en tabel, som indeholder alle data for de to tidligere tabeller.



Nu indeholder spørgsmålstabellen alle spørgsmål og alle svar. den reserverer altså plads i databasen til alle spørgsmål og alle svar. 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.

Derfor ændres tabellerne endnu engang, og spørgsmålstabellen får lov at stå alene, mens svar tabellen refereres direkte til tjeklistetabellen. Det er en én



til én relation mellem tjekliste og svar tabel, så derfor inkluderes kolonnerne fra svar tabellen direkte i tjeklistetabellen.

