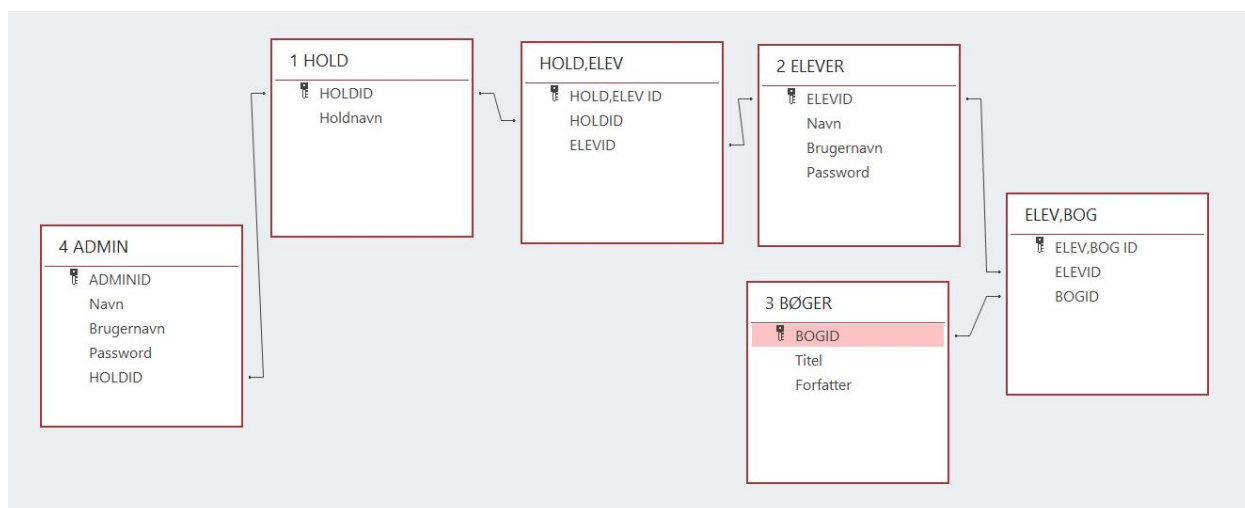


DatabaseBibliotek - overvejelser

Opgaven

Vores oprindelige tanke med opgaven var, at vi ville lave et HTX-bibliotek system, hvor der findes to typer brugere: elever og adminer/lærere. Elever skulle kunne låne bøger og se de bøger, som du har udlånt. Administratorer skulle kunne se sine hold med holdets elever og de bøger, som eleverne har udlånt. Rent kodningsmæssigt ville dette betyde, at vi skulle lave en database med fire tabeller (admin; elever; bøger; hold) og to mellemtabeller for at kunne lave mange-mange relationer mellem data i tabellerne (hold-elever; elever-bøger). Vores relationer mellem tabellerne kan ses på *figur 1*. Herudover var vores mål, at systemet skulle kunne nulstilles, så databasen slettes, og der skulle kunne indlæses nye elever/klasser fra Lectio. Adgangskoderne skulle være krypteret eller hashet, og der skulle være en backup-fil af databasen.

Herudover havde vi en idé om, at eleverne og admin skulle kunne se, hvor lang tid bøgerne havde været udlånt, hvis vi fik tid til overs til at lave dette.



Figur 1. Her ses vores relationer mellem tabeller (lavet i Access). Der er 4 tabeller: Hold, elever, bøger og admin. Der er 2 mellemtabeller, som forbinder henholdsvis hold-elever og elever-bøger.

User story

Til opgaven havde vi skrevet følgende user story:

Eleverne skal kunne åbne en fil, der refererer til en biblioteksdatabase, så eleverne kan se deres udlån. Eleverne skal kunne aflevere og låne bøger. Der skal kunne være en administrator, som adgang til alle elevernes profiler. Administrator skal kunne slette alle lånte bøger for en elev/klasse/hold. Der skal være en backup fil.

Det endelige resultat

I vores endelige kode har vi lavet og brugt den database, som vi har beskrevet, at vi først havde tænkt, vi ville lave under afsnittet "Opgaven". Vi lavede en oversigt/skitse over databasen i Access, hvorefter vi skrev databasen i SQLite.

Vi har lavet biblioteket-systemet således, at man logger ind som bruger på startsiden; man indtaster sit brugernavn og sit password, hvorefter man trykker på login-knappen. Her har vi valgt at gå efter et traditionelt login-side design ved at bruge webkonventioner, hvilket gør programmet mere brugervenligt. Vi har valgt at bruge biblioteket "ControlP5" i Processing til at lave tekstboksene, hvor brugeren skriver sit brugernavn og password, da biblioteket har en del features, hvilket gør mulighederne for at videreudvikle på programmet senere hen større. Hvis brugeren taster et forkert brugernavn og/eller password, vil tekstbokserne blive ryddet, og brugeren vil **ikke** blive logget ind.

Når brugeren er logget ind, vil den næste side være forskellig alt efter, om det er en elev eller admin, der logger ind. En admin ville kunne se alle elever samt deres udlånte bøgers titel og forfatter, mens en elev kun ville kunne se de bøger, som eleven selv har udlånt (titel + forfatter).

Måden hvorpå en bruger logger ind er indtil videre, at brugernavn er de fire første bogstaver i brugerens navn, og adgangskoden er det første bogstav i brugerens fornavn skrevet tre gange med store bogstaver. Altså ville Mads' brugernavn og password som et eksempel være:

Brugernavn: Mads

Password: MMM

Af brugere har vi for eksempel oprettet (brugere som man kan logge ind med):

Elever:

Cecilia

Elisabeth

Jessie

Mads

Emil

Nicoline

Kaja

Admin:

John

Anders

Perspektivering

Hvis vi havde mere tid til at lave på programmet, ville den første ting, vi ville have arbejdet på, være at hashe passwordet, som brugeren bruger til at logge ind med. Grunden til, at vi ville hashe passwordet og ikke kryptere det, er fordi vi ikke er interesseret i at kunne kende det egentlige password igen senere hen; vi er kun interesseret i at vide, om brugeren har tastet det rigtige password. Derfor vil kryptering ikke være nødvendigt.

Derudover ville vi gerne have lavet password feltet således, at tegnene kunne ses som prikker eller stjerner i stedet for de tegn, som brugeren skriver, så en anden person ikke ville kunne kigge brugeren over skulderen og se brugerens password. Vi ville desuden gerne have lavet en scroll-funktionen inde på selve elevens- og admins side, hvilket er nødvendigt, da det ellers er begrænset, hvor mange bøger (og for admins tilfælde elever), som brugeren vil kunne se.

Til sidst ville vi gerne have lavet en nulstillings-funktion, så databasen ville kunne blive slettet, og vi ville gerne have gjort det muligt at skrive nye elever/klasser ind i systemet. Herudover ville vi gerne have haft en ekstra kolonne på admins- og elevens side, så man kunne se, hvornår eleven havde udlånt bogen (og eventuelt endnu en kolonne, hvor man ville kunne se, hvor lang tid bogen havde været udlånt).