11.12.2018

Gruppe 5

Navn: Joakim Westby Studentnr: 218165 Email: 218165@student.usn.no

Navn: William Rastad Studentnr: 216411 Email: 216411@student.usn.no

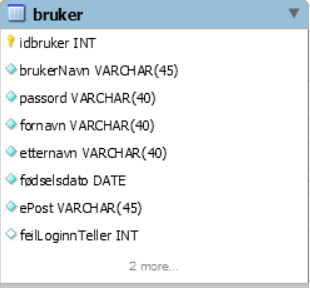
Navn: Kristoffer Sørensen Studentnr: 216387 Email: 216387@student.usn.no

Navn: Simen A. Lyse Studentnr: 216608 Email: 216608@student.usn.no

**Databasemodellen**

Vi har lagt til fornavn, etternavn og fødselsdato i tillegg til databasemodellen i vedlegg 1. Dette er på grunn av profilsiden, hvor vi ønsker å kunne sette navn på profilene så brukeren kan kommunisere med hverandre og vite hvem som er hvem. brukerNavn = studentnummer.

Oppdatert databasemodell:



**Registrering**

**Login**

For innlogging har det blitt brukt PDO for tilkobling mot databasen.

For å motvirke sql-injection har det blitt brukt prepared statements. Dette er en form for kommunikasjon der query parameterene blir sendt sepparat fra sql-spørring inn til databasen(Huseby, 2004, s.39). Sql-injekson vil derfor ikke være mulig og "vasking" av escape characters ikke være nødvendig i denne sammenhengen.

En registrert brukers passord vil være lagret som en hash i databasen som er opprettet med krypteringsalgoritmen sha1 kombinert med foranstillt salt "IT2\_2019". Dette gjøres to ganger for å øke sikring mot angrep ved bruk av rainbow tables.

sha1(salt + ( sha1(salt + passord)))

Logg-inn fungerer, men vi har ikke inkludert forsvarsmekanismer mot bruteforce angrep.

Den planlagte løsningen:

|  |  |  |
| --- | --- | --- |
| feilLoginnSiste: | feilLoginnTeller:1 or 2 | feilLoginnTeller: +=1  forsøk gjennomføres |
| feilLoginnSiste: | feilLoginnTeller:0 | feilLoginnTeller: += 1  feilLoginnSiste: cur.tid  forsøk gjennomføres |
| feilLoginnSiste:<5m siden | feilLoginnTeller:3 | bruker må vente |
| feilLoginnSiste:>=5m siden | feilLoginnTeller:3 | feilLoginnSiste: cur.tid  feilLoginnTeller: 1  forsøk gjennomføres |

Referanser

Huseby, H, H. (2004). Innocent code. West Sussex: John Wiley & Sons, Ltd.