**Asignment 3 – NoSQL**

**Gruppe 6**

**Jeppe Dybdal Larsen au616966 201805466**

**Mikkel Kousgaard Rasmussen au610713 201805606**

I denne opgave med MongoDb har vi valgt at designe vores system således at vi kun arbejder på en collection, nemlig brugere (Users). En brugere har en række opslag (posts) associeret med sig – Disse er embedded i vores bruger collection. Et opslag har dog stadig et unik ID sat til sig, da vi løbende ønsker at arbejde på et specifik opslag – F.eks. når en kommentar skal skrives til et opslag. Kommentarer er yderligere embedded i en post. Post holder altså en liste af kommentarer i sig, på samme måde som hver bruger holder en liste af post og en liste af circles.

Nedenfor er vist et diagram, der med inspiration fra et ER diagram, forsøger at give et overblik over det designede system. Det er dog vigtigt at huske, at alt vist i diagrammet nedenfor er indeholdt i et enkelt dokument i Users collection.

Et billede, der indeholder tekst, kort

Automatisk genereret beskrivelse

**Overvejelser**

Vores design endte med at bestå af kun en collection – Users – Hvor alt andet interessant information er embedded i bruger dokumenterne. Grunden bag dette, var at vi ønskede at nedsætte tiden det tog at hente information fra databasen. Da alt relevant information for en specifik bruger allerede er en del af dokumentet (f.eks. Posts og Circles), kræver det kun at finde det ene dokument fra databasen. Med denne løsning kommer dokumenterne godt nok til at fylde mere og mere, og det er ikke nødvendigvis holdbart, hvis en bruger begynder at slå flere tusinde opslag op, hvilket kan resultere i at bruger-dokumentet kommer til at fylde for meget ift. MongoDbs maksimum størrelse for dokumenter.

Man kunne have overvejet for både opslag og cirkler, at have dem i en collection for sig selv. Med ordentlige shard keys og indeksering kunne man stadig få et hurtigt system til at indhente både opslag og cirkler for en bestemt bruger. Ved at dele cirkler, brugere og opslag op i forskellige collections, løber man dog risikoen af at skulle gennemsøge alle cirkler/opslag for alle registrerede brugere, for at kunne lave et feed/wall for en bestemt bruger. Dette var netop grunden til at vi endte med at embedde disse til en bruger.

**Sharding**

I forhold til horisontal skalering af databasen, ville en åbenlys shard key være UserId’et – Altså det objekt id som dokumentet bliver tildelt ved indsættelse i databasen. Det er netop dette Id der bliver queriet på, når der ønskes at indlæse en væg og et feed for en person i vores sociale netværk. En anden mulighed for shard key kunne være, hvis brugeren også havde oplyst informationer om hvor han/hun havde bopæl. Denne adresse/by kunne derved være med til bestemmelse af shard key, så dokumentet for brugeren blev lagt på en passende database, der lå så tæt på byen/landet som muligt.

**Indexing**

Da vores sytemt udelukkende querier på brugeren/dokumentets ID, ville det være oplagt at indekserer ud fra dette.