Dokumentasjon – Treningsdagbok

Tabellkorresponderende klasser:

De tabell-korresponderende klassene validerer om unike rader allerede finnes i databasen før de sender SQL-spørring til databaseserveren, og returnerer det nylige genererte elementet som et klasseobjekt. Slik håndteres SQL-feil i applikasjonen.

Apparat.java

Registrerer et fysisk treningsapparat, med en beskrivelse om hvordan det fungerer.

Bruker.java

Registrerer en bruker med brukernavn og alder.

Applikasjonen skal sannsynligvis benyttes av svært mange brukere, og dermed har vi nytte av å ha et brukernavn som fremmednøkkel i «treningsøkt»-tabellen.

Notat.java

Registrerer et notat opp mot en gitt treningsøkt ut ifra treningsøktID som kan inneholde informasjon om personlig form, prestasjon, treningsformål og opplevelse av treningsøkten. Det er frivillig å registrere et notat til en treningsøkt.

Ovelse.java

Registrerer diverse øvelser en bruker kan utføre ved navn og beskrivelse. Dekker muligheten for øvelser med og uten apparater.

Dersom øvelsen ikke benytter et apparat, vil feltet «apparat» være tom og «harApparat» være lik *false*.
Dersom øvelsen benytter et apparat vil feltet «apparat» peke til et «Apparat»-objekt, og «harApparat» være lik *true*.

Slik implementerer vi arv-relasjonen som vi har beskrevet i diagrammet vårt. Subklassene til superklassen Ovelse er Ovelse med apparat og Ovelse uten apparat.

OvelseGruppe.java

Registrerer en øvelsesgruppe med øvelsesgruppenavn.

OvelsegruppeTilOvelse.java

Registrerer en relasjon mellom en øvelsesgruppe og en enkel øvelse. Dette gjøres med et unikt par av øvelsesgruppenavn og øvelsesnavn, nemlig at begge inngår i tabellens primærnøkkel.

Treningsokt.java

Registrerer en treningsøkt, med bruker, dato, tidspunkt og varighet. I *Notat.java* er det mulighet for å registrere tilleggsinformasjon om økten, dersom brukeren ønsker det.

TreningsoktTilOvelse.java

Registrerer en relasjon mellom en treningsøkt og en enkel øvelse, med antall kilo og sett som ble utført. Dette gjøres med et unikt par av treningsøktID og øvelsesnavn, nemlig at begge inngår i tabellens primærnøkkel. Antall kilo og sett registreres som null dersom øvelsen ikke benytter et apparat.

Hovedkontroller-klasse:

Treningsapp.java

Inneholder main-metode, kjører og orkestrerer applikasjonen med klasser.

Klassen kobler seg opp mot databasen, og tilbyr brukeren funksjonalitet gjennom et tekstlig grensesnitt og tilstandsautomat. Alle tidligere nevnte klasser benyttes for å løse sine relevante oppgaver, og bidrar til å løse kravene til funksjonalitet.

Use-cases:

1. *«Registrere apparater, øvelser og treningsøkter med tilhørende data.»*

   Brukeren kan enkelt registrere apparater, øvelser og tilhørende data gjennom metodene i applikasjonen. Brukeren skriver inn data etter oppfordring, og det lagres til databasen gjennom de tabellbaserte klassene.

2. *«Få opp informasjon om et antall* **n** *sist gjennomførte treningsøkter med notater, der* **n** *spesifiseres av brukeren.»*

Siden Notat- og Treningsøkt-tabellene er knyttet sammen i databasen ved fremmednøkkel treningsøktID, vil vi kun trenge å lete gjennom Notat-tabellen og hente radene fra Treningsøkt med korresponderende treningsøktID.
Dette er implementert med et inndata-felt etter å ha valgt bruker.

3. *«For hver enkelt øvelse skal det være mulig å se en resultatlogg i et gitt tidsintervall spesifisert av brukeren.»*

   Gjennom å slå sammen tabellene TreningsoktTilOvelse og Treningsokt kan vi få ut resultatene (antall kilo og sett) til en øvelse i et ønsket tidsintervall gjennom å kjøre en query på den sammenslåtte tabellen.

4. *«Lage øvelsegrupper og finne øvelser som er i samme gruppe.»*

   Hver øvelse er underlagt en øvelsesgruppe gjennom en relasjon. Dersom vi ønsker å hente ut øvelser fra samme gruppe, trenger vi kun å filtrere ut ØvelsesgruppeTilØvelse-tabellen ut i fra øvelsegruppeNavn, og se på øvelsesnavnene. Applikasjonen vil returnere navnene på øvelser som er tilknyttet øvelsesgruppen.

5. *«Et use-case for denne databasen kan være en privatperson som ønsker å ha oversikt over treningsprogrammet sitt, eller en personlig trener som ønsker å loggføre treningene til kundene sine.»*

   En privatperson ser de ulike tilgjengelige øvelsene, og ved å benytte applikasjonen til sitt fulleste, så kan hen holde rede på hvilke øvelser de utfører ved å sortere TreningsøktTilØvelse-tabellen ut i fra treningsøktID, til hvilke tidspunkter ved Treningsøkt-tabellen ut i fra treningsøktID og hvordan de presterer ved Notat-tabellen ut i fra treningsøktID.
   Dette gjøres i applikasjonen ved å f.eks velge ut treningsøkter, se på øvelsesgrupper osv.

   En personlig trener kan filtrere Treningsøkt-tabellen ut ifra brukernavn, og dermed få en liste over alle utførte treningsøkter under det gitte navnet.
   Dette gjøres i applikasjonen ved å velge den ønskede brukeren, og be om en liste av treningsøkter. Treneren kan følge prosessen i forrige avsnitt for å holde rede på sine klienters fremgang. Det er også mulig å se innskrevede notater.