

## 1 Dokumentasjon av DB2

Vi har valgt å gå for en fat-controller løsning hvor vi har en omtrentlig mapping mellom en tabel og en javaklasse. Alle kontroller klasser spesifiserer også en getAll-metode for å hente ut alle objekter av denne typen fra databasen.

## 2 Oversikt over klasser

### 2.1 DBConnect.java

Er en abstrakt klasse som håndterer oppkobling mellom kontrollerobjekter og databasen.

### 2.2 DeviceCtrl.java

Håndterer registrering og muligheten til å hente ut apparater i databasen. Usecase 1.

### 2.3 ExerciseCtrl.java

Håndterer registrering av øvelser og muligheten til å hente ut disse. Registrering returnere id på øvelsen slik at en mapping mellom øvelsesgruppe og øvelse blir muliggjort. Usecase 1.

### 2.4 ExerciseGroupCtrl.java

Håndterer forholdet mellom en øvelse og øvelsesgruppen den tilhører. Dette innebærer registrering av øvelsesgrupper, mapping mellom øvelse og øvelsesgruppe, og muligheten til å hente ut øvelser i samme øvelsesgruppe. Usecase 4.

### 2.5 ExerciseInWorkoutCtrl.java

Håndterer forholdet mellom øvelse og treningsøkt. Under registrering av treningsøkt kan man registrere øvelser tilhørende økta. Her skjer altså registrering av forhold. Total statistikk kan også returneres fra objektet. Samt en opplisting av øvelser utført i et gitt tidsintervall. Usecase 3, Usecase 5.

### 2.6 Workout.java

Et treningsøktobjekt som validerer input fra brukeren og holder på denne dataen. Usecase 1.

## 2.7 WorkoutCtrl.java

Kontroller som styrer lagring og uthenting av treningsøkter. Her kan man både registrer og hente ut n tidligere treningsøkter. Registrering returnerer id-en på treningsøktobjektet for å muliggjøre relasjonen mellom øvelser og treningsøkter. Ved registrering av flere øvelser på samme treningsøkt tar ExerciseInWorkoutCtrl seg av en del av arbeidet. Usecase 1, Usecase 2.

## 2.8 Main.java

Klassen som interagerer med brukeren og holder oversikten over hva som skal gjøres. Som nevnt i neste hovedpunkt har vi benyttet en slags hovedmeny løsning hvor bukeren starter med flere valg, utfører ett av disse, og sendes deretter tilbake til hovedmenyen. Mer utdypet om programmets helhet under punktet "Oversikt over løste usecase".

## 3 Oversikt over løste usecase

Klassene i punktet over er tagget med usecase de hovedsaklig har som oppgave å løse. I programmet forgår mye av løsningen direkte gjennom reg- og get-metoder som i seg selv er ganske selvforklarende. Main-metoden kjører ulike deler av programmet gjennom dedikerte menyfunksjoner slik som createExercise(), createDevice() o.l. Vi har partisjonert programmet inn i en meny som muliggjør returnering til velkommen-funksjonen etter at ønsket aksjon er utført. Velkommen-funksjonen er da en hovedmeny som lar brukern velge mellom all funksjonalitet spesifisert av usecasene. Det valgfrie usecase 5, som omhandler total statistikk, er løst slik at bruker ved start av programmet får opp total statistikk basert på resultatene som tidligere er lagt inn. Ellers finnes det som sagt dedikerte metoder som løser spesifikk usecase direkte. Om ønskelig vil en kjapp titt på Main.java sammen med metodenavn og oversikten over gjør den spesifikke kode-implementasjonen synlig.