

Dokumentasjon innlevering 2

Av Kristoffer Rakstad Solberg, Sigmund Hennum Høeg, Petter Sletsjøe

1. Klasser og beskrivelse av hvilke oppgaver hver klasse skal løse:

DatabaseConnectivity:

Denne klassen sørger for at man kan få kontakt med MySQL-databasen. Dette gjøres ved å kalle funksjonen *openConnection()* som oppretter forbindelse til databasen. Etter det er opprettet kontakt med databasen kan man utføre spørringer med følgende funksjoner: *executeGetQuery()* som sørger tar i mot en spørring av typen `SELECT * FROM <TABLE>`. Klassen inneholder også funksjon for å sette inn i databasen, dette gjøres med *executeSetQuery()* som tar inn spørringer av typen `INSERT INTO <TABLE> VALUES()`. Etter spørringer er utført vil klassen sørge for å lukke forbindelsen til MySQL-databasen ved å kjøre funksjonen *closeConnection()*.

MainController:

Denne klassen styrer logikken i sammenheng med brukeropplevelsen. Den fungerer som en automaton som tar imot input fra brukeren og responderer på denne. Den tar inn info fra brukeren om nye øvelser, grupper, økter og apparater som skal registreres i databasen. Controlleren prøver også å lage objekter utifra informasjonen gitt av brukeren, og sender dette til slutt til Handler-klassen som tar seg av spørringer opp mot DatabaseConnectivity for å lagre informasjonen der.

ControllerState:

Håndterer enums i forhold til hvilken spørring man ønsker å utføre

Exercise, Equipment, Workout:

Enkle dataklasser med konstruktører.

Handler:

Handler-klassen tar seg av logikk som vi vil holde utenfor MainController.

2. Oversikt over use cases som er løst og hvordan de er realisert i programmet:

1. Register an exercise
 - `INSERT INTO Exercise(name, description) VALUES (" + exercise.name + ", " + exercise.description + ");`
2. Register a workout

- ***"INSERT INTO Workout VALUES ('" + workout.name + "', '" + workout.description + "');"***;
3. Register an equipment
 - ***"INSERT INTO Equipment (name, description) VALUES ('" + equipment.name + "', '" + equipment.description + "');"***;
 4. Register an exercise group
 - ***"INSERT INTO ExerciseGroup VALUES (" + name + ", " + description + ");"***;
 5. Show all exercises in the database
 - ***SELECT * FROM Exercise;***
 6. Show all workouts in the database
 - ***SELECT * FROM Workout;***
 7. Show all equipment in the database
 - ***SELECT * FROM Equipment;***
 8. Show all exercise groups in the database
 - ***SELECT * FROM ExerciseGroup;***
 9. Show all exercises in a group
 - ***SELECT Exercise.exerciseID, Exercise.name, Exercise.description FROM Exercise JOIN ExerciseInGroup NATURAL JOIN ExerciseGroup WHERE ExerciseGroup.exerciseGroupID = 2;***
 10. See the n last workouts with notes
 - ***SELECT Workout.workoutID, Workout.note FROM Workout LIMIT 0,n;***
 11. See the results of one exercise in a given time interval
 - ***SELECT * FROM ExerciseInWorkout NATURAL JOIN Workout WHERE Workout.workoutDate > DATE ('2019-06-06') AND Workout.workoutDate < DATE('2019-08-06');***
 12. See the most frequent exercise
 - ***SELECT ExerciseID, count(*) as count FROM ExerciseInWorkout GROUP BY ExerciseID ORDER BY count DESC LIMIT 1;***