

# Opdracht programmeren gevorderd 2

## Context

Fitness “YouMove” wenst de service naar hun klanten te verbeteren en heeft daarom gevraagd om een applicatie te ontwikkelen voor het beheer van klanten, fitnessstoestellen en trainingsdata.

Voor elke klant houden ze zowel de voornaam als achternaam bij, het emailadres, hun verblijfplaats, geboortedatum, een lijst met interesses en het type klant. Klanten worden ingedeeld op basis van hun inschrijving en ontvangen dan het label Bronze, Silver of Gold. De lijst met interesses is optioneel, alle andere velden zijn verplicht. Wat betreft de verblijfplaats is de naam van de gemeente voldoende.

Klanten kunnen zich ook inschrijven voor een programma dat in de fitness wordt aangeboden. Een programma heeft een unieke code, een naam, een doelpubliek, startdatum en een maximum aantal leden dat kan deelnemen aan het programma.

De fitness beschikt ook over een aantal toestellen, voorlopig enkel loopbanden en fietsen die elk een unieke id hebben en een korte beschrijving van het type toestel.

Klanten kunnen deze toestellen reserveren, daarbij geven ze aan welk toestel ze wensen en wanneer (dag en tijdslot) dat ze dat wensen te gebruiken. Een tijdslot is telkens 1 uur en bij een reservatie kan 1 specifiek toestel maximaal 2 tijdsloten na elkaar worden gereserveerd. Je kan in totaal ook maximaal 4 tijdsloten per dag gebruiken om te reserveren. We onderscheiden ochtendsessies tussen 8 en 12 (dus 4 slots), middagsessies vanaf 12h tot 18h en avondsessies van 18h tot 22h. In totaal zijn er dus 14 slots op 1 dag.

Een reservatie bevat de volgende info :

- klantnummer, emailadres, voor en achternaam
- datum
- voor elk tijdslot het gereserveerde toestel

Zorg er ook voor dat er enkel reservaties kunnen plaatsvinden in de toekomst (lijkt me evident) en maximaal 1 week op voorhand.

Het kan voorkomen dat een toestel stuk gaat en er onderhoud nodig is, dan moet dat toestel tijdelijk niet beschikbaar worden. In dit geval kunnen de bestaande reservaties niet doorgaan, stel een gebruiksvriendelijke oplossing voor !

Er kunnen ook nieuwe toestellen geleverd worden en die moeten dan ook aan de lijst met te reserveren toestellen worden toegevoegd. Als we een toestel verwijderen uit de fitness dan moeten wel nog alle reservaties uit het verleden beschikbaar blijven. We verwijderen trouwens geen toestellen die nog reservaties hebben.

Opmerking :

- Unieke codes voor de toestellen moeten door de databank worden aangemaakt.

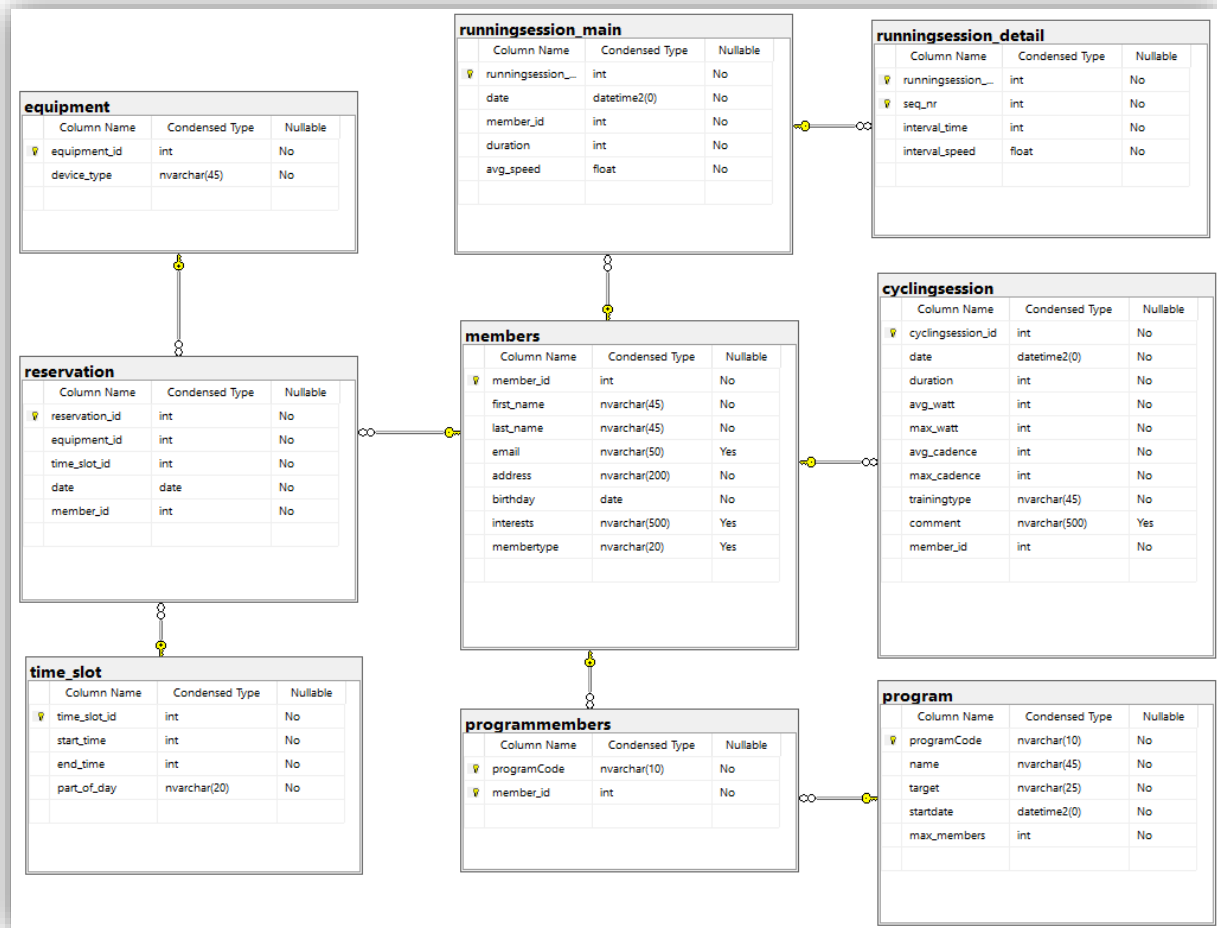
- Klantnummers moeten door de databank worden aangemaakt.
- Ook reservatienummers moeten door het systeem worden aangemaakt.

Momenteel is er een bestaande databank met daarin zowel gegevens van de klanten, toestellen, programma's en reservaties samen met gegevens over trainingssessies. Er zijn twee verschillende trainingssessies, één voor fietsen en één voor lopen. Voor een fietssessie wordt de totale duur in minuten opgeslagen, samen met de gemiddelde en maximale wattage, gemiddelde en maximale cadans en het traingstype (opties zijn fun, endurance, interval en recovery). Daarnaast is er ook een mogelijkheid om commentaar mee te geven. Voor een looptraining maken we gebruik van geavanceerde toestellen waarbij de snelheid wordt bijgehouden telkens de gebruiker deze aanpast. Naast de klantnummer wordt ook de datum en starttijd bijgehouden, de gemiddelde snelheid en de totale looptijd. Daarnaast wordt voor elk interval de tijdsduur en de snelheid gedurende dit tijdsinterval opgeslagen. In het voorbeeld is dit duidelijk te zien, de looper heeft zijn loopsessie opgedeeld in 4 intervallen, een eerste interval van iets meer dan 20 min aan een snelheid van 20 km/h, daarna 20 min rustig aan (10 km/h), weer een snel interval aan 19 km/h en een laatste 20 minuten aan ongeveer 15 km/h.



## Databank

Er is reeds een databank beschikbaar en het schema ziet er als volgt uit :



Er is ook een script beschikbaar met SQL-statements om de verschillende tabellen aan te maken en de gegevens te importeren. Indien de import correct is verlopen zou je over de volgende gegevens moeten beschikken.

SELECT count(*) FROM [GymTest].[dbo].[cyclingsession]
SELECT count(*) FROM [GymTest].[dbo].[equipment]
SELECT count(*) FROM [GymTest].[dbo].[members]
SELECT count(*) FROM [GymTest].[dbo].[programmamembers]
SELECT count(*) FROM [GymTest].[dbo].[reservation]
SELECT count(*) FROM [GymTest].[dbo].[runningession_detail]
SELECT count(*) FROM [GymTest].[dbo].[runningession_main]
SELECT count(*) FROM [GymTest].[dbo].[time_slot]

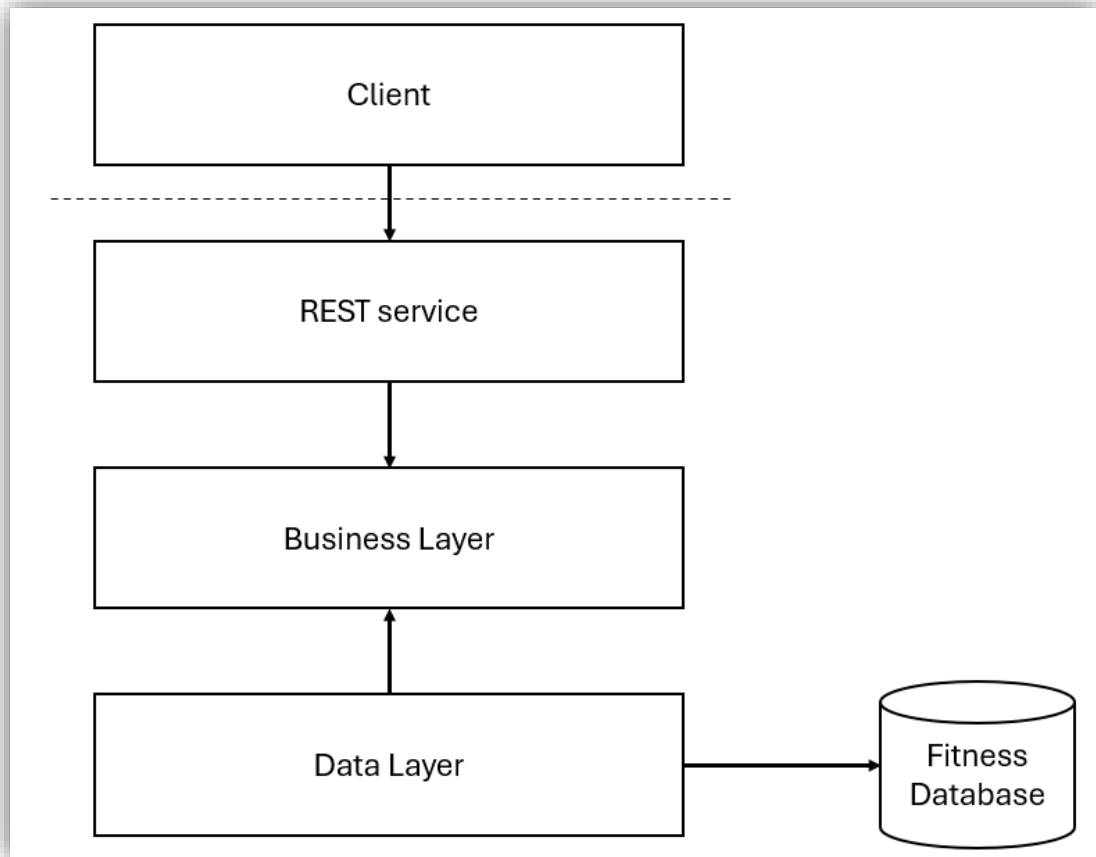
  

Results	Messages
(No column name)	
50722	
(No column name)	
12	
(No column name)	
500	
(No column name)	
330	
(No column name)	
4092	
(No column name)	
169208	
(No column name)	
21907	
(No column name)	
14	

## Architectuur

Er moet gewerkt worden in een gelaagde architectuur waarbij de business-laag centraal staat. De data laag moet gemaakt worden op basis van Entity Framework, waarbij we voorzien dat een omschakeling naar een ander databank-systeem eenvoudig kan plaatsvinden zonder al teveel impact op de andere lagen.

Voor de bouw van de REST-service maken we gebruik van ASP.NET.



## REST-service

Er moet een REST-service gebouwd worden die de volgende functionaliteiten voorziet :

- Nieuwe klanten toevoegen
- Gegevens van de klant aanpassen
- Toestellen toevoegen
- Toestellen in onderhoud plaatsen
- Reservaties aanmaken
- Reservaties verwijderen
- Reservaties aanpassen
- Programma toevoegen
- Programma wijzigen
- Gegevens van klant opvragen
  - Algemene gegevens
  - Reservaties
  - Programma's waarvoor ingeschreven
  - Overzicht van de trainingssessies
- Voorzie de mogelijkheid om gedetailleerde gegevens van een trainingssessie op te vragen
- Voorzie de mogelijkheid om statistische gegevens op te vragen

- Geef alle sessies voor een klant voor een bepaalde maand en jaar, gesorteerd op datum.
- Geef het totaal aantal sessies, tijdsduur in uren, langste en kortste sessie en gemiddelde tijdsduur van een sessie voor een klant.
- Geef voor een bepaalde klant een overzicht van het aantal sessies per maand voor een specifiek jaar.
- Idem als voorgaande maar maak verder onderscheid per trainingstype.
- Voeg voor elke sessie de trainingsimpact (een definitie van trainingsimpact wordt nog aangeleverd) toe aan het maandoverzicht.

### ***Trainingsimpact***

*Het laatste verzoek is om voor elke sessie de trainingsimpact toe te voegen aan het maandoverzicht. De trainingsimpact zal bepaald worden op basis van de trainingsduur en de geleverde inspanning. Indien het gemiddelde wattage van een sessie kleiner is dan 150 en de sessie duurde maximaal anderhalf uur dan beschouwen we de impact als laag ('low'). Is het wattage kleiner dan 150, maar duurde de training langer dan 90 minuten dan stellen we dat de impact gemiddeld is of 'medium'. Sessies met wattages tussen de 150 en 200 beschouwen we altijd als 'medium' en sessies met wattages boven de 200 watt categoriseren we als hoog ('high').*

### **Opmerkingen :**

Als er aanpassingen aan de databank nodig zijn, dan kan je die gerust uitvoeren.

## **Client-Applicatie**

Bouw een applicatie waarmee op een gebruiksvriendelijk manier een reservatie kan worden aangemaakt. De voorkeur gaat uit om deze applicatie in WPF te maken, optioneel kan er ook gebruik gemaakt worden van React.

## **Opdracht**

- Bouw een REST service die voldoet aan de gestelde eisen.
  - Bezorg een overzicht van de service door middel van Swagger.
  - Zorg dat er een demo kan worden gegeven op basis van Swagger.
- Bouw een gebruiksvriendelijk client-applicatie (WPF) waarmee we een reservatie kunnen aanmaken. Zorg er ook hier voor dat er een demo kan worden gegeven.
- Voorzie de nodige unit tests (Xunit) voor de service (unit tests voor de business-laag zijn niet nodig).
- De code moet worden beheert via git.

Veel success !