



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di Laurea Magistrale in Informatica

Assignment 3

Processo e Sviluppo del Software

Andrea Carubelli 803192

Gianmaria Balducci 807141

Link repository:

https://gitlab.com/jiimmy.exe/2019_assignment3_gym_sharing/

Anno Accademico 2019-2020

Overview dell'applicazione

Lo scopo dell'applicazione è quello di permettere agli utenti di cercare nella propria zona le palestre disponibili, consultando gli abbonamenti erogati, con i relativi prezzi e servizi offerti.

Inoltre è possibile che un utente scelga un personal trainer da cui può essere seguito durante i suoi allenamenti.

E' possibile inoltre per ogni utente, condividere il proprio abbonamento con altri utenti, così da poter ammortizzare il costo dello stesso.

Diagramma E-R

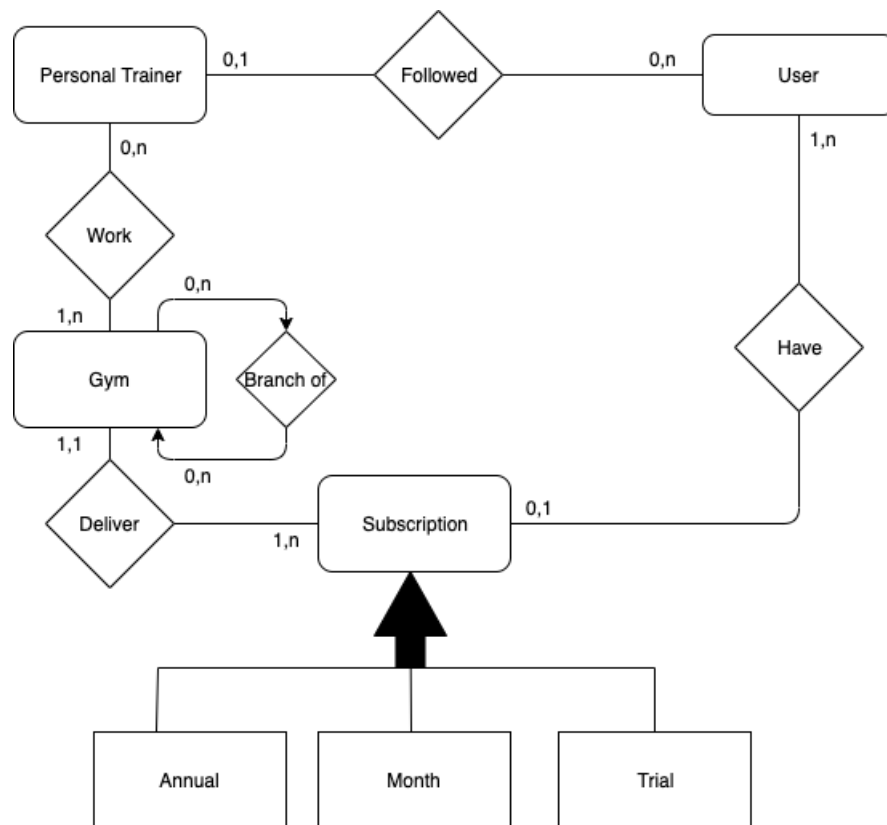


Figura 1: Diagramma E-R

Dal diagramma E-R possiamo osservarne le relazioni:

- Un utente può possedere un solo abbonamento, ma un abbonamento può essere condiviso fra più utenti
- Una palestra può erogare più abbonamenti ma un abbonamento può essere erogato da una sola palestra
- Una palestra può avere delle filiali attraverso una relazione "self-loop"

- In una palestra ci possono essere più personal trainer e un personal trainer può lavorare in più palestre
- Un utente può essere seguito da uno o nessun personal trainer e un personal trainer può seguire più utenti

Diagramma UML

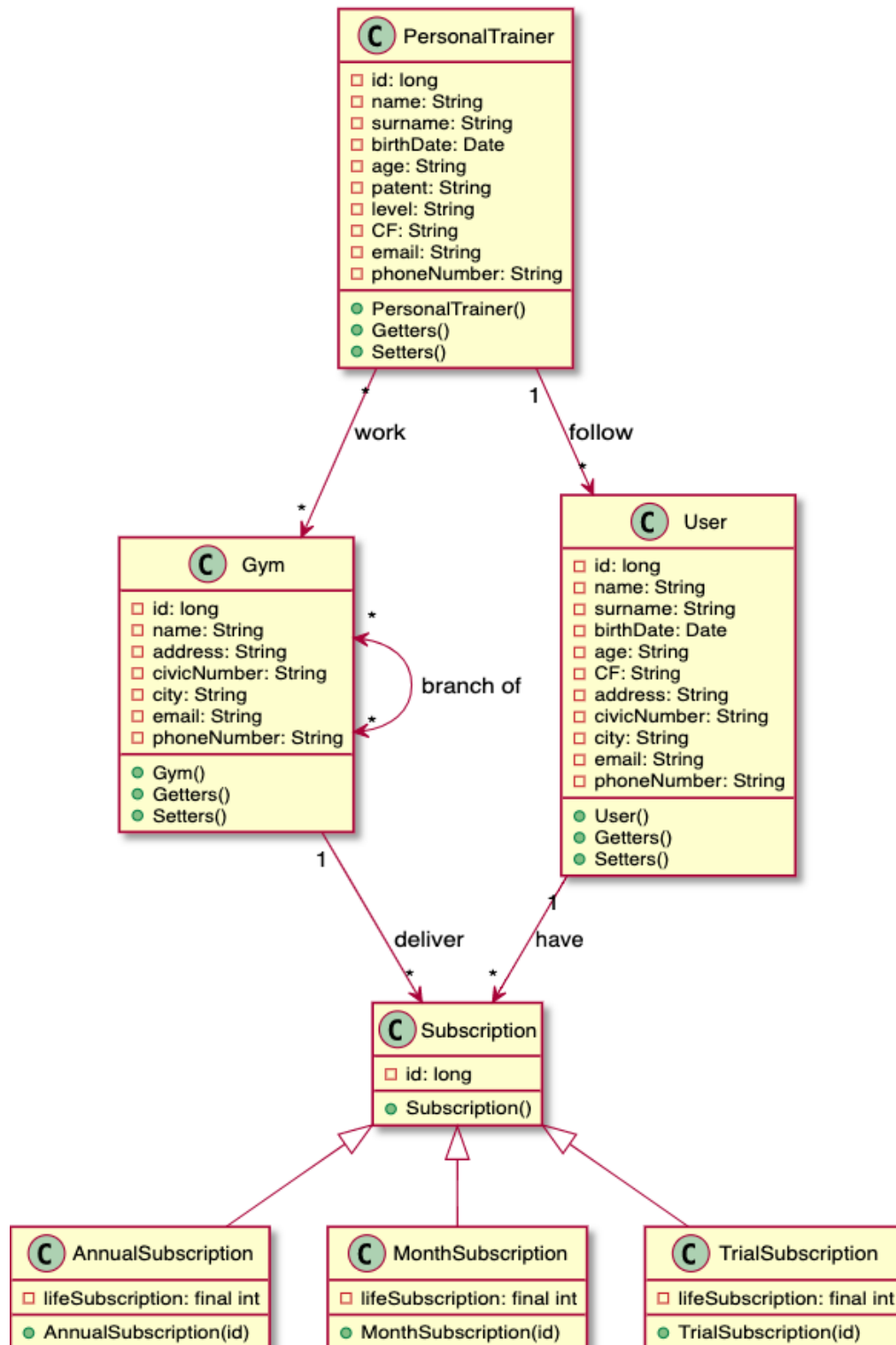


Figura 2: Diagramma UML

Descrizione classi

Abbiamo scelto l'implementazione "front-end only" per un' applicazione Spring-MVC.

Il nostro package base è `com.example.assignment3` in cui sono inclusi i seguenti package:

- **Entities**
- **Controller**
- **Repository**

Il package **Entities** contiene le seguenti entità:

- **Gym** che rappresenta le palestre
- **User** che rappresenta gli utenti
- **Personal Trainer** che rappresenta i personal trainer
- **Subscription** che rappresenta gli abbonamenti erogati dalle palestre. **AnnualSubscription**, **MonthSubscription**, **TrialSubscription** ereditano questa classe

Il package **Controller** contiene i controller, uno per ogni classe sopracitata che permetteranno alla webapp di chiamare le operazioni CRUD da effettuare e muoversi tra le views. I controller presenti, per adesso, sono i seguenti:

- **BasicController** che gestisce le *route* delle views.
- **GymController** che gestisce le operazioni CRUD per la classe Gym
- **UserController** che gestisce le operazioni CRUD per la classe Gym

Il package **Repository** contiene le interfacce che ci permetteranno di salvare gli oggetti in modo volatile.

Al momento, le interfacce create sono:

- **GymRepository**
- **UserRepository**