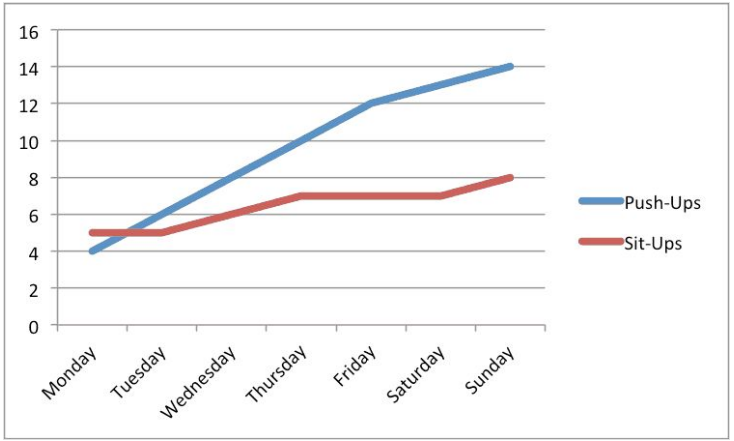
myFitness Analytics

Alberto Valle & Shivneel Chand

Github repo: https://github.com/Betovef/fitness-tracker

System Implementation

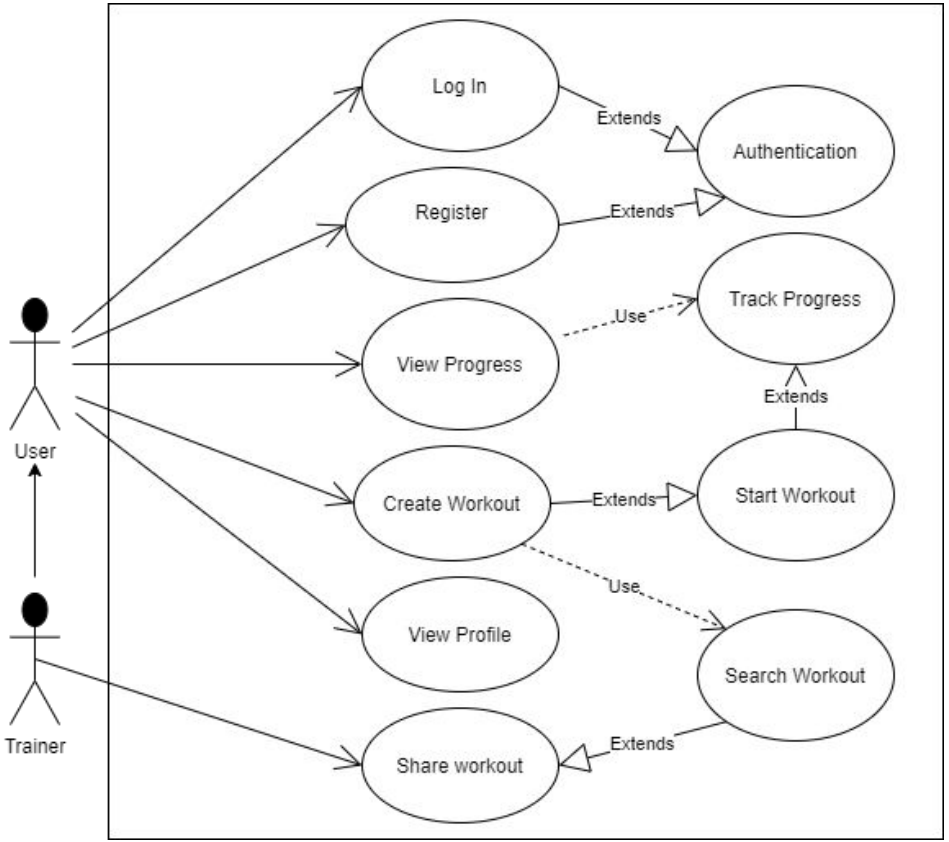
- Fitness app
- Users will be able to:
  - Search a database for different exercises
  - Add new workouts
  - View their progress
  - Subscribe to trainers to view their workouts



Github repo: https://github.com/Betovef/fitness-tracker

UML

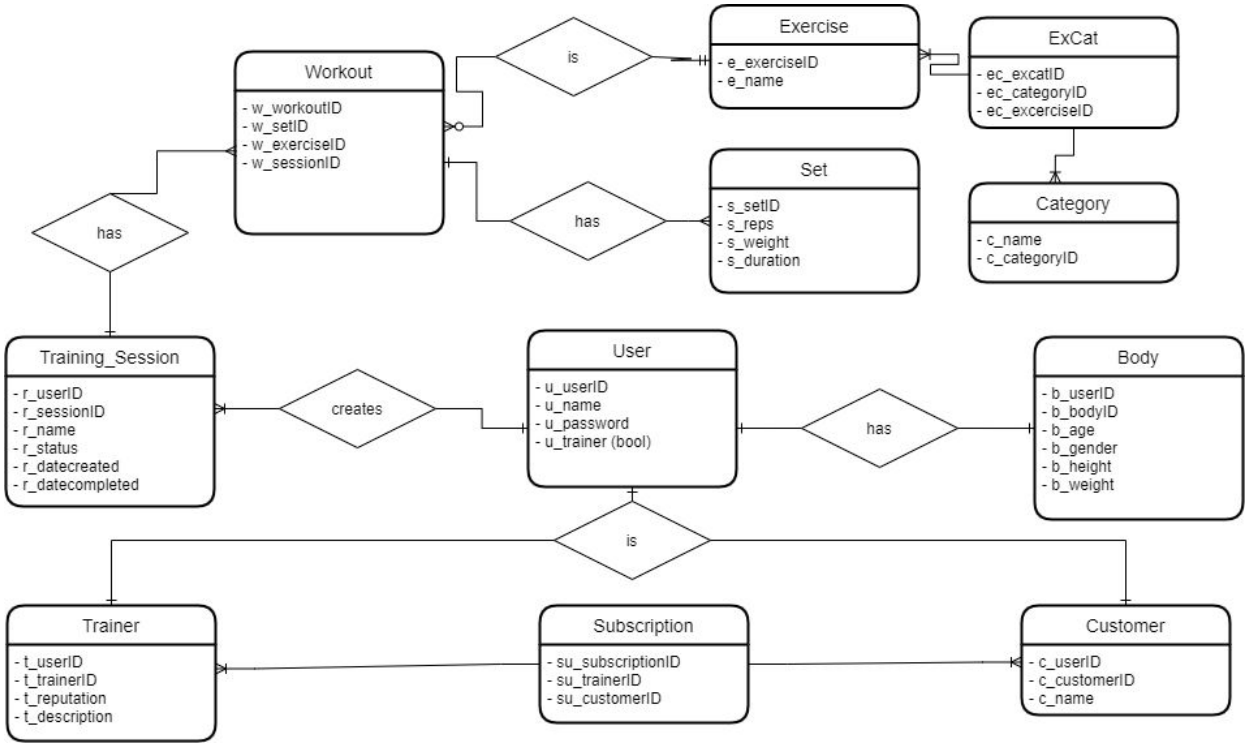
- User sessions
  - Registration/Login
- Create Workout
  - Choose exercise
  - Add sets / reps
- View Progress
  - Search workout by date
  - Line graph
- Subscriptions
  - Trainer and Customer
  - View Trainer workouts



Github repo: https://github.com/Betovef/fitness-tracker

E/R Diagram

- 9 Entities
- One User has one Body and is a Customer and/or Trainer
- One User creates Many Training_Sessions
- One Training_Session has many Workouts
- One Workout has many Sets
- Many Workouts are one Exercise
- Many Exercise can have Many Categories



Github repo: https://github.com/Betovef/fitness-tracker

Relational Schema

- User
  - u_userID, u_name
    u_password, u_trainer (bool)
- Body
  - b_userID, b_bodyID, b_age,
    b_gender, b_height, b_weight
- Customer
  - c_userID, c_customerID
- Trainer
  - t_trainerID, t_userID, t_reputation,
    t_description
- Subscription
  - su_subID, su_customerID,
    su_trainerID
- Workout
  - w_workoutID, w_setID, w_exerciseID,
    w_sessionID
- Exercise
  - e_exerciseID, e_name
- Set
  - s_setID, s_reps, s_weight, s_duration
- Category
  - c_name, c_categoryID
- Training_Session
  - r_userID, r_sessionID, r_name, r_status,
    r_datecreatedm, r_datecompleted
- ExCat (connects category to exercise)
  ec_id, ec_categoryID, ec_exerciseID,

Github repo: https://github.com/Betovef/fitness-tracker

Implementation Details

Web Application:

- Backend
  - Python Flask
- Database
  - SQLite
- Frontend
  - HTML
  - CSS
  - JS
- Graphics and Interactivity
  - D3.js