**Workout Tracker**

User Interface

* Main window(Profile) with a calender visualisation of exercises + 4 buttons

(Profile, History, Workout, Exercises)

* History window where you can see and edit past workouts
* Workout window where you can make a new workout
* Exercises window where you can see, add, and edit exercises

Business Logic Tier

* Workout class
  + WorkoutID
  + WorkoutDate(DateTime)
  + WorkoutName(string)
  + WorkoutNote(string)
  + List<Exercise>
* Exercise class
  + ExerciseName
  + Sets(int)
  + Reps(int)
  + Weigth(double)
  + //WeigthUnit(string)
  + RPE(int)
* AddClass
* EditClass
* DeleteClass

Data Tier

* WorkoutMapper
  + GetWorkoutFromDB
  + AddExerciseToWorkout
  + DeleteWorkoutFromDB
* ExerciseMapper
  + GetExercisesFromDB
  + GetExerciseByNameFromDB
  + AddExerciseToDB
  + DeleteExerciseFromDB

Requirements per workout

* Date(DateTime)
* Workout name(String)
* Add exercises per workout(class)(minimum 1)
* Add sets per exercise(int)(minimum 1)
* Add reps per exercise(int)(minimum 1)
* Weigth(Double)
* Weigth unit(String)
* RPE(int)(1-10 | default = 0)
* Add notes per exercise and per workout(string)

Using in Angular:  
HammerJS