# General-Application Description

The desired application is intended for users who wish to develop and track a workout routine while also needing access to many customization options. These customization options extend beyond just tracking the weight or time of different exercises, but also for tracking differences in a given workout's reps, time-to-completion, total-weight-lifted, etc. The application should also allow the user to add or customize exercises. These exercises should allow extensive information descriptions to help a user to, for example, rapidly swap a regimen’s exercises with others targeting similar muscle groups.

In terms of tracking exercises, users should be able to enter relevant information for both short- and long-term analysis. In the short-term, a user might want to track the number of lifts per repetition, the weight lifted, or other metrics both during and post-workout. For long-term tracking, it should be possible to arrange the data from individual work-out logs into different “viewables”, such as charts or calendrical arrangements (heat maps, etc.). The purpose of this long-term tracking doubles to provide the user a visual representation of their accomplishments, as well as help a user to discover patterns with their regimen like, for example, if it doesn’t include many exercises targeting specific muscle groups.

The application should also provide an avenue for the user to plan out their future workouts or regimens. The user can choose a future date (or recurring timestep/week/day of the week) to dedicate to some user-defined workout session. These sessions are designed around exercises which are predefined by the user as discussed above, but are also finalized in this stage, when the user can enter the weight being lifted in a rep, the number of lifts they intend to accomplish (or to continue until X lifts before failure), etc. The user can also order the exercises during planning so that during the workout they can verify their plan at a glance. These regimens should be saveable, and the saved regimens editable, so that the user can easily assign the same regimen to any additional day (while changing target weight loads, etc.).

The final core feature will be the ability for the user to upload their data to an external database. The user should be able to store data locally for ease-of-use during workout, however, to prevent data loss and limit the application’s local storage space requirements, the user should be able to upload their data to the database. The **user** should also be able to retrieve data from the database, in the case where the user begins using a new phone/computer. For this reason, the database would typically be up-to-date or behind the locally-stored data, so if the user tries to pull from the database while the local data is “ahead” of the uploaded-data, it should prompt the user if they wish to rollback to the stored or cancel the pull.

Potential features include the ability to define stretches separately to exercises, in which the user can add a user-designed stretch-routine before/after a target exercise regimen with just a click.

Highlight legend:

Feature

Ill-defined potential feature

Definitely a class/activity

Might be a class/activity

Not a class/repeat of existing class

| Class | Variables/Methods | Notes |
| --- | --- | --- |
| Exercise | name: String  description: String  tags: Tag[]  thumbnailID: String  exerciseStatistics: ExerciseStats[] | thumbnailID can be linked to a proper image by whatever activity might be displaying the exercise, as to limit storage space |
| Tag | name: String  taggedExercises: Exercise[] | Reference class, so if the user wants to display all exercises with tag X, it can simply use the taggedExercises list |
| WorkoutLog | date: Date  timeCompleted: Time  regimen: Regimen  exerciseStats: ExerciseStats[] |  |
| Regimen | name: String  regimenExercises: Exercise[] | This holds an user-ordered list of exercises, to help structure a user’s workout plans |
| ExerciseStats | exercise: Exercise  partOfWorkout: WorkoutLog  metric: METRIC\_STATE  metricValue:  repValues: ArrayList<double> | metric will define what values are being tracked. Generally, an exercise is analyzed by tracking one value relative to another:   * # of lifts at a weight, * the weight given a # of lifts * time to achieve a distance * distance in a given time * etc.   For this reason,  repValues is appended with whatever value was achieved for the chosen metric during repetitions 1, 2, etc.  double chosen for repValues as it should hold any timing edge cases, while allowing conversion to integers for metrics like number of presses |

Bench Press: lifts @ weight

Running: distance @ time or time @ distance (time & distance → speed)

Plank: time held @ weight, weight @ time held