UserAdministration

+ AddUser(user: User): void
+ DeleteUser(username: string): void
+ EditPassword(username: string, newPassword: string): void
+ ValidateUserIsUnique(username: string): bool
+ GetUser(username: string): User
+ ChangePassword(user: User, newPassword: string): void

— 1 — HAS — 1..* →

User

- firstName: string
- username: string
- password: string

+ FirstName{get; set}: string
+ Username{get; set}: string
+ Password{get; set}: string
+ User(name: string, username: string, password: string)

CardioAdministration

+ AddCardio(name: string, calories: int, difficulty: string)
+ GetCardio(name: string): Cardio

— 1 — HAS — 1.. * →

Cardio

- name: string
- calories: int
- difficulty: string

+ Name{get;set}: string
+ Calories{get;set}: int
+ Difficulty{get;set}: string
+ Cardio(name: string, calories: int, difficulty: string)
+ ToString(): string

<<interface>>
IAdministration

+ ValidateExerciseIsUnique(name: string): bool
+ ExerciseExists(name: string): bool
+ RemoveExercise(name: string): void

StorageManager

+ LoadUsers(myManager: UserAdministration): UserAdministration
+ LoadExercises(myManager: ExerciseAdministration): ExerciseAdministration
+ LoadCardios(myManager: CardioAdministration): CardioAdministration

ExerciseAdministration

+ ExerciseAdministration()
+ CreateExercise(typeEx: string, nameEx: string, difficulty: string, equipment: string, numReps: int, weight: double, specialty: string): void
+ AddExercise(newEx: Exercise): void
+ GetExercises(): Exercise[]
+ GetExercise(name: string): Exercise
+ EditExercise(exName: string, reps: int, amount: double, difficulty: string): void
+ GetExercisesList() : List<Exercise>
+ EditExerciseData(ex: Exercise, newReps: int, newWeight: double, difficulty: string): void
+ AddDummyData(ex: Exercise): void

— 1 — HAS — 1..* →

Exercise

- name: string
- difficulty: string
- equipment: string
- repRange: int
- weight: double
- picture: string

+ Name{get; set}: string
+ Difficulty{get; set}: string
+ Equipment{get; set}: string
+ RepRange{get; set}: int
+ Weight{get; set}: double
+ Exercise(name: string, difficulty: string, equipment: string, repRange: int, weight: double, picture: string)
+ToString(): string
+ CompareTo(ex: Exercise): int

<<interface>>
IComparable<T>

+ CompareTo(other: T): int

CardioDAL

+ CreateCardio(): void
+ UpdateCardio(): void
+ DeleteCardio(): void
+ GetAllCardio(): Cardio[]

ExerciseDAL

+ CreateExercise(): void
+ UpdateExercise(): void
+ GetAllExercises(): Exercise[]
+ DeleteExercise(): void

UserDAL

+ CreateUser(): void
+ UpdateUser(): void
+ DeleteUser(): void
+ GetAllUsers(): User[]

BaseDAL

- connectionString: string

BaseDAL()
CreateConnection(): MySqlConnection

ShoulderExercise

- shoulderHead: string

+ ToString(): string
+ ShoulderExercise(name: string, difficulty: string, equipment: string, reps: int, weight: double, picture: string, shoulderHead: string)

NeckExercise

+ ToString(): string
+ NeckExercise(name: string, difficulty: string, equipment: string, reps: int, weight: double, picture: string)

ChestExercise

- chestElement: string

+ ToString(): string
+ ChestExercise(name: string, difficulty: string, equipment: string, reps: int, weight: double, chestElement: string, picture: string)

LegsExercise

- legs: string

+ ToString(): string
+ LegsExercise(name: string, difficulty: string, equipment: string, reps: int, weight: double, legs: string, picture: string)

ArmExercise

- armElement: string

+ ToString(): string
+ ArmExercise(name: string, difficulty: string, equipment: string, reps: int, weight: double, armElement: string, picture: string)

BackExercise

- backElement: string

+ ToString(): string
+ BackExercise(name: string, difficulty: string, equipment: string, reps: int, weight: double, backElement: string, picture: string)

CoreExercise

+ ToString(): string
+ CoreExercise(name: string, difficulty: string, equipment: string, reps: int, weight: double, picture: string)