GymSystemDao «interface»

+persist(object: Object): void
+remove(object: Object): void
+close(object: Object): void
+update(object: Object): void
+query(query: String, parameters: Map<String, Object>): List
+querySingle(query: String, parameters: Map<String, Object>): Object

EntityManagerDao

+persist(object: Object): void
+remove(object: Object): void
+close(object: Object): void
+query(query: String, parameters: Map<String, Object>): List
+querySingle(query: String, parameters: Map<String, Object>): Object

GymSystemHandler
+GymSystemHandler()

SystemHandler
+SystemHandler()

TrainerHandler
+TrainerHandler()

ManagerHandler
+ManagerHandler()

CustomerHandler
+CustomerHandler()

LoginHandler
+LoginHandler()

GymSystemHandler
+GymSystemHandler()

PersonalInformationBuilder
#firstName: String
#lastName: String
#phoneNumber: String
#email: String
#street: String
#city: String
#state: String
#zipcode: String
#healthInsurance: String

+PersonalInformationBuilder()
#createPersonalInformation(): PersonalInformation
#updatePersonalInformation(info: PersonalInformation): void

GymSystemUserBuilder
#username: String
#password: String
#confirmPassword: String
#id: int

#createUser(userLevel: String): User
#updateGymSystemUser(user: GymSystemUser): GymSystemUser

ManagerBuilder
+createManager(): Manager

TrainerBuilder
+createTrainer(): Trainer
+updateTrainer(trainerId: int): Trainer

CustomerBuilder
-membershipStatus: String

+CustomerBuilder()
+createCustomer(): Customer
+updateCustomer(customerId: int): Customer

SystemHandler
+SystemHandler()
+getHealthInsurances(): List<HealthInsurance>
+getAllStates(): List<State>
+getInventory(): List<Machine>
+searchInventory(): List<Machine>

HealthInsurance
- name: String

+ HealthInsurance(name: String)

State
- state: String
- stateAbbrev: String

- State(state: String, stateAbbrev: String)

ManagerHandler
ManagerHandler

+createManager(email: String, firstName: String, lastName:String, phoneNumber:String, insurance: String, userName: String, userPass: String, confirmPass: String, street:String, city: String, zip: String, state: String): Manager
+getAllManagers(): List<Manager>
+searchManager(keywords: String)
+getManagerById(id: String):
+addMachine(name: string: image: String, quantity: String): Machine
+updateMachine(name: string: image: String, quantity: String): Machine
+removeMachine(id: String, uploadPath: String): Machine
+getMachineById(id: String): Machine

Manager

Machine
- name: String
- picture: String
- quantity: int

+ Machine(name: String, picturePath:String, quantity: int)

TrainerHandler
TrainerHandler()

+createExercise(name: String, machineId, timeDuration:int, minDuration: int, secDuration:int, noOfSets: int, noOfRepetition: int): Exercise
+modifyExercise(name: String, machineId: String, timeDuration:String, minDuration: String, secDuration:String, noOfSets: String, noOfRepetition: String): Exercise
+deleteExercise(exerciseId: int): void
+searchExercises(value: String): List<Exercise>
+getAllExercises(value: String): List<Exercise>
+getExerciseById(exerciseId: int): Exercise
+createTrainer(firstName: String, lastName:String, phoneNumber:String, email:String, street:String, city: String, state: String, zipcode: String, healthInsurance: String, userName: String, password: String): Trainer
+modifyTrainer(firstName: String, lastName:String, phoneNumber:String, email:String, street:String, city: String, state: String, zipcode: String, healthInsurance: String, userName: String, password: String): Trainer
+deleteTrainer(id: int): void
+getTrainerById(trainerId: int): Trainer
+getTrainerByUserid(userId: int): Trainer
+searchTrainers(value: String): List<Trainer>
+getAllTrainers(): List<Trainer>
+addQualification(trainerId: int, qualification: String): void
+deleteQualification(trainerId: int, qualification: String): void
+addWorkHours(trainerId: int, startDateTime: DateTime, endDateTime): void
+deleteWorkHours(trainerId: int, workHoursId: int): void
+getWorkHourById(workHourId: int): WorkHour
+createWorkout(name: String, exerciseNames: String[]): Workout
+updateWorkout(workoutId: String, name: String, exerciseNames: String[]): Workout
+searchWorkouts(keyword: String): List<Workout>
+deleteWorkout(workoutId: String): void
+assignWorkout(customerId: int, workoutId: int): void
+unassignWorkout(customerId: int, workoutId: int): void
+getWorkoutsByCustomerId(customerId: int): List<Workout>

WorkHours
- startDateTime: Date
- endDateTime: Date
/ days: List<String>
/ hours: int
/ minutes: int

+ WorkHours(startDateTime: Date, endDateTime: Date)

Qualification
- name: String

+ Qualification(name: String)

Trainer
+ Trainer(personalInformation: PersonalInformation, user: User, workInformation: WorkInformation)
+ addWorkHours(wh: WorkHour): void
+ deleteWorkHours(): void
+ addQualification(qualification: Qualification): void

Exercise
- name: String

+ Exercise(name: String)
+ deleteDuration(): void
+ addSet(exerciseSet: ExerciseSet): void
+ deleteSet(exerciseSet: ExerciseSet): void
+ deleteSets(): void

Workout
- name: String

+ Workout(name: String, trainer: Trainer)

CustomerHandler
+ CustomerHandler()

+createNewCustomer(email: String, firstName: String, lastName:String, phoneNumber:String, insurance: String, street:String, city: String, zip: String, state: String, membershipStatus: String): Customer
+updateCustomer(id: int, email: String, firstName: String, lastName:String, phoneNumber:String, insurance: String, street:String, city: String, zip: String, state: String, membershipStatus: String): Customer
+removeCustomer(id: String): void
+getCustomers(): List<Customer>
+getCustomerByKeyword(keyword: String): List<Customer>
+getCustomerById(id: int): Customers

Customer
+Customer(personalInformation: Personal Information membership: Membership)
+ addWorkout(workout: Workout): void
+ removeWorkout(workout: Workout): void
+ removeAllWorkouts(): void

LoginHandler
+LoginHandler()
+authenticate(username: String, pass: String): boolean
+getUserByUserName(userName: String): User
+getUserLevel(username: String): String
+getUserId(userName: String): int

User
- username: String
- password: String

+ User(username: String, password: String, level: UserLevel)

UserLevel
- description: String

+ UserLevel(description: String)

Text