

«boundary» view::MainMenu
-id: int «PK» -stage : Stage -profilePanel : SubScene -trainingPanel : SubScene
+show() : void

«controller» controller::MainController
-currentUser: User -views: int[]
+startApp() : void +openMainMenu() : void +openProfileView() : void +logout() : void

«controller» controller::TrainingController
-currentUser: User -views: int[]
+recommendExercises() : List<Exercise> +createPlan() : Plan +createDay() : Day

«boundary» view::ProfileView
-id: int «PK» -stage : Stage -user : User -progressChart : LineChart
+showUserData() : void +updateProfileData(user : User) : void

«entity» model::User
-id : int «PK» -username : String -age : int -weight : double -height : double -sex : String -password: String -plans: Plan[] -days: Days[]
+delete():void +synchronize():void +updateProfile(age : int, weight : double,...) : void

«entity» model::Exercise
-id : int «PK» -name : String -description : String -muscleGroup : String -difficulty : String -duration : int
+isSuitable(user : User) : boolean +toString() : String

«boundary» view::Panel
-root : Pane -title : String
+render() : void +updateContent() : void

«entity» model::Plan
-id: int «PK» -name: String -startDate: date -endDate: date -days: Day[]
+addDay():void +deleteDay():void +updateDates(newStartDate : date, newEndDate : date):void

«entity» model::Weight_Exercise

«entity» model::Cardio_Exercise

«entity» model::Day
-id: int «PK» -date: date -steps: int -caloriesIntake : int -caloriesBurned : int -exercises: Exercise[] -exercises_completed: boolean
+addExercise(newExercise : Exercise): void +deleteExercise(deleteEx : Exercise): void +getNetCalories() : int +updateSteps(steps : int) : void