* **Class and Method Documentation**: Descriptions of classes, methods, and their purposes.

Изображение выглядит как текст, снимок экрана, Шрифт, Параллельный

Автоматически созданное описание

**The classes roles and interactions:**

1. **CalorieCalculator Class**:
   * **Role**: Calculates calorie needs based on user data.
   * **Methods**:
     + calculateBMR(): Calculates Basal Metabolic Rate.
     + calculateCalories(): Computes total daily calorie needs.
     + caloriesForWeightLoss(), caloriesForMuscleGain(), caloriesForMaintenance(): Provides calories required for different goals.
     + getDisclaimer(): Returns a disclaimer with sources.
2. **MacroCalculator Class**:
   * **Role**: Calculates macronutrient needs based on user goals.
   * **Methods**:
     + calculateBMR(): Calculates Basal Metabolic Rate (could be reused from CalorieCalculator).
     + calculateCalories(): Computes total daily calorie needs (could be reused from CalorieCalculator).
     + calculateMacros(): Computes macronutrient distribution for different objectives (maintenance, weight loss, muscle gain).
     + getDisclaimer(): Returns a disclaimer with sources.
3. **Profile Class**:
   * **Role**: Represents user profile data.
   * **Methods**:
     + Getters and setters for profile attributes like weight, height, age, activity\_level, desired\_objective.
4. **Database\_connection Class**:
   * **Role**: Manages database connection.
   * **Methods**:
     + getConnection(): Returns the PDO database connection instance.
5. **Repository Class**:
   * **Role**: Handles database operations related to User, Profile, and Statistics.
   * **Methods**:
     + fetchAllUsers(), fetchAllUserNames(), fetchAllProfiles(), fetchAllStatistics(): Fetch data from corresponding tables.
     + getUser($user\_id): Fetch user by ID.
     + createUser(User $new\_user): Create a new user.
     + getUserId($email): Get user ID by email.
     + getUserByCredential($credential): Get user by username or email.
     + updateUser($email, $updated\_data): Update user details.
     + deleteUser($user\_id): Delete user by ID.
     + getUserProfile($user\_id): Get user profile by user ID.
     + createProfile($user\_id, Profile $new\_profile): Create a new profile.
     + updateProfile($user\_id, $updated\_data): Update user profile.
     + updateObjective($user\_id, $desired\_objective): Update the desired objective in the profile.
     + addProgress($user\_id, $progressData): Add progress data to the statistics table.
     + getUserStatistics($user\_id): Get user statistics.
6. **User Class**:
   * **Role**: Represents user data.
   * **Methods**:
     + Getters and setters for user attributes like username, password, email, first\_name, last\_name.

**Class Relationships and Flow:**

1. **CalorieCalculator and MacroCalculator**:
   * **Relationship**: Both classes are related to calculating nutritional needs but are independent. MacroCalculator could potentially use CalorieCalculator for its calorie calculations.
   * **Interaction**: MacroCalculator might use CalorieCalculator methods for calorie calculations if needed, or you can refactor common methods into a base class.
2. **Profile and Repository**:
   * **Relationship**: Profile is used to manage user profile data, and Repository handles database operations for Profile.
   * **Interaction**: Repository methods interact with Profile instances to store and retrieve profile data.
3. **Database\_connection and Repository**:
   * **Relationship**: Database\_connection provides the database connection to Repository.
   * **Interaction**: Repository uses Database\_connection to execute database queries.
4. **User and Repository**:
   * **Relationship**: User represents the user data, and Repository manages database operations related to User.
   * **Interaction**: Repository uses User instances to perform create, read, update, and delete operations on user data.