

DietBase - Personal Diet and Nutrition Tracker

By: Zach Angha

Student ID: 922988765

GitHub Username: zachangha

Version History:

Checkpoint#	Date Submitted
Checkpoint I	02/20/24
Checkpoint II	3/5/24
Checkpoint III	4/1/24

Table of Contents

<i>Product Description.....</i>	<i>3</i>
<i>Functional Database Requirements.....</i>	<i>6</i>
<i>Non-functional Requirements</i>	<i>11</i>
<i>Entity Relationship Diagram.....</i>	<i>13</i>
<i>Entity Description.....</i>	<i>14</i>
<i>Entity Establishment Relationship Diagram.....</i>	<i>22</i>
<i>Constraints Description</i>	<i>23</i>

Product Description

Ever since I have been going to the gym, I have found using a personal diet and nutrition tracker app to be very beneficial to make sure I am getting all the right nutrients. Since I am lifting multiple times a week and looking to gain muscle, I have tried to make sure I am in a caloric surplus and eating a high-protein diet. On the other hand, I have many friends who use these apps for the opposite reasons to lose weight, meaning they want to be in a caloric deficit. This shows that these apps can be used for both people trying to bulk up and people who are aiming to lose weight. As someone who has been using many nutrition-tracking apps for the past couple of years, I understand all the standard features that come with them, but all pretty much have the same functionality and features. For these companies to separate themselves from the competition, they could look to implement a new database system that can open the gate for new features that would attract more consumers.

The DietBase database system is perfect for helping people with their busy lives, assisting them plan meals or to just track what they have available to them, while offering the same features offered by any other diet or nutrition tracker. With the current dieting, tracking calories and nutrients has become very easy, but what if you want more precise tracking where you can input what you buy from the store like a virtual fridge or pantry? DietBase will allow more precise tracking of food by implementing a tracking table where we will store all the foods and ingredients that people buy from the store collecting what groceries they currently have all in one place. We offer this database to dieting software and apps that want to add new and innovative features for their users, to improve their customer satisfaction, saving them time and energy. What separates DietBase from other dieting databases is that we will allow users to input and track what they have in their fridge or pantry for possible recipes, track freshness dates, or just take inventory of what they currently have. Features like alerts of when you run out of an ingredient and adding it directly to your shopping list, notifying you of expired foods, or recommending recipes that you can make with what you have can all be added with this new tracker. Existing software like MyFitnessPal and WeightWatchers would greatly benefit from this feature because they'd be able to offer their customers something that would make their apps and systems more personal, allowing them to track their groceries through the app.

Use Cases:

1. **Use Case:** Food Waste Reduction

Actor: Environmentalist (Steven)

Description: Steven is very concerned about the environment and reducing food waste is one of the many ways for him to do his part in saving it. Steven will use anything and everything he buys to prevent the waste in resources, food, and money. Steven wants to have a way to track the expiration date of all the foods he buys from the grocery store, so he can be sure nothing he buys goes to waste in his house.

DietBase is the solution to his problem as it has tables made to track all the foods you buy from the store and all the expiration dates, so you can view how close the food gets to losing its freshness. Apps could implement features with this database that will let Steven view expiration dates and be notified when his food is getting close to becoming expired.

2. **Use Case:** Limited Meals

Actor: Late Night Worker (Josh)

Description: Josh works late shifts at his job and when he gets home, he just wants to eat something before he goes to sleep. All the restaurants near Josh are closed by the time he gets off, so he's limited to what he has at home. Sometimes Josh doesn't have many ingredients and is restricted by what he must work with.

DietBase can help him find recipes that fit within his constrictions. The DietBase database system saves the ingredients in recipes as singular food entities, allowing apps to develop a system that will query and find recipes that he can make with what he has available.

3. **Use Case:** Grocery List

Actor: Full-time Student (Jessica)

Description: Jessica is a full-time student and has class almost every day of the week. She doesn't have convenient access to a grocery store, so she must make sure she gets everything she needs, the one time she goes a week.

The DietBase system is a solution to this problem as there is a built-in grocery list table. Developers could implement the feature for Jessica to let her create her own grocery list in the app, so she never forgets anything again.

Functional Database Requirements

1. Users: Strong
 - 1.1. A user shall create one account.
 - 1.2. A user shall login to many devices.
2. Devices: Strong
 - 2.1. A device shall log into many users.
3. Account: Weak
 - 3.1. An account shall be created by at only one user.
 - 3.2. An account shall be a general user or an admin.
4. General User: Weak
 - 4.1. A general user shall have at least one allergies and restrictions.
 - 4.2. A general user shall have one account setting entry.
 - 4.3. A general user shall save only one measurement.
 - 4.4. A general user shall have at least one long-term goal.
 - 4.5. A general user shall have only one set of nutritional goals.
 - 4.6. A general user shall input at least one weight log.
 - 4.7. A general user shall input at least one fasting log.
 - 4.8. A general user shall input at least one water intake.
 - 4.9. A general user shall have at least one step counter.
 - 4.10. A general user shall input at least one sleep log.
 - 4.11. A general user shall create at least one recipe, food, medication, or supplement.
 - 4.12. A general user shall input at least one recipe, food, medication, or supplement into a diet log.
 - 4.13. A general user shall save at least one recipe.
 - 4.14. A general user shall create at least one exercise.
 - 4.15. A general user shall input at least one exercise in their exercise log.
 - 4.16. A general user shall have at least one meal plan.
 - 4.17. A general user shall put at least one food to their grocery list.
 - 4.18. A general user shall put at least one food to their food inventory.

- 4.19. A general user shall add many friends with other accounts.
- 4.20. A general user shall join at least one group.
- 4.21. A general user shall create at least one post.
- 4.22. A general user shall be sent at least one notification.
- 4.23. A general user shall accept at least one challenge.
- 5. Admin: Weak
 - 5.1. An admin shall do everything a general user can do.
 - 5.2. An admin shall delete posts.
 - 5.3. An admin shall delete comments.
 - 5.4. An admin shall delete accounts.
 - 5.5. An admin shall delete recipes.
 - 5.6. An admin shall delete foods.
- 6. Allergies and Restrictions: Strong
 - 6.1. An allergy and/or restriction shall be assigned to many accounts.
- 7. Account Settings: Weak
 - 7.1. An account settings entry shall be assigned to only one account.
 - 7.2. An account settings entry shall have one and only one language selected.
 - 7.3. An account settings entry shall have one and only one location selected.
 - 7.4. An account settings entry shall have one and only one time zone selected.
 - 7.5. An account settings entry shall have one and only one measurement system selected.
- 8. Time Zones: Strong
 - 8.1. A time zone shall be selected by many account settings.
- 9. Locations: Strong
 - 9.1. A location shall be selected by many account settings.
- 10. Languages: Strong
 - 10.1. A language shall be selected by many account settings.
- 11. Measurement Systems: Strong
 - 11.1. A measurement system shall be selected by many account settings.

12. Measurements: Weak

12.1. A measurement shall be assigned to only one account.

13. Long-Term Goals: Weak

13.1. A long-term shall be assigned to only one account.

14. Nutritional Goals: Weak

14.1. A set of nutritional goals shall be assigned to only one account.

15. Weight Log: Weak

15.1. A weight log shall be input by only one account.

16. Fasting Log: Weak

16.1. A fasting log shall be input by only one account.

17. Water Intake: Weak

17.1. A water intake shall be input by only one account.

18. Step Counter: Weak

18.1. A step counter shall be assigned to only one account.

19. Sleep Log: Weak

19.1. A sleep log shall be input by only one account.

20. Food: Strong

20.1. A food shall be created by at most one account.

20.2. A food shall be input to many diet logs.

20.3. A food shall have at most one complete nutritional information.

20.4. A food shall be used in many recipes.

20.5. A food shall be in many grocery lists.

20.6. A food shall be in many food inventories.

20.7. A food shall be assigned many food types.

21. Medications: Strong

21.1. A medication shall be created by at most one account.

21.2. A medication shall have at most one complete nutritional information.

21.3. A medication shall be input to many diet logs.

22. Supplements: Strong

- 22.1. A supplement shall be created by at most one account.
- 22.2. A supplement shall be input to many diet logs.
- 22.3. A supplement shall have at most one complete nutritional information.

23. Recipes: Weak

- 23.1. A recipe shall have at least one food.
- 23.2. A recipe shall be created by at most one account.
- 23.3. A recipe shall be input to many diet logs.
- 23.4. A recipe shall have at most one nutritional information.
- 23.5. A recipe shall be saved by many accounts.
- 23.6. A recipe shall be used in many meal plans.
- 23.7. A recipe shall be created by at most one account.
- 23.8. A recipe shall be assigned at least one meal category.
- 23.9. A recipe shall be assigned many meal types.

24. Exercises: Strong

- 24.1. An exercise shall be created by at most one account.
- 24.2. An exercise shall be input by many accounts.

25. Meal Plans: Weak

- 25.1. A meal plan shall be assigned to at least one account.

26. Groups: Strong

- 26.1. A group shall have many accounts.

27. Posts: Strong

- 27.1. A post shall be created by at most one account.

28. Notification: Weak

- 28.1. A notification shall be sent to an account.

29. Challenges: Strong

- 29.1. A food challenge shall be accepted by many accounts.

30. Nutritional Information: Weak

- 30.1. A nutritional information shall be assigned to only one food, recipe, supplement, or medication.

31. Meal Category: Strong

31.1. A meal category shall be assigned to many recipes.

32. Food type: Strong

32.1. A food type shall be assigned to many foods.

Non-functional Requirements

1. Performance

- 1.1. The database shall support concurrent inputs.
- 1.2. The database shall respond to the user's input in a reasonable amount of time.
- 1.3. The database shall be optimized to find the needed information in a reasonable amount of time.
- 1.4. The database shall take in the user's information in a reasonable amount of time.
- 1.5. The database shall handle concurrent user activity without having a significant impact on the performance.

2. Security

- 2.1. The database shall protect the user's measurements from other people.
- 2.2. The database shall only store encrypted passwords.
- 2.3. The database shall be backed up every day at 11:59 pm.
- 2.4. The database shall only insert values that are consistent with the attribute's datatype and domain.
- 2.5. Administrative functions and commands are only given to those that are authorized to do so.

3. Scalability

- 3.1. Regardless of the number of users the database shall function as expected.
- 3.2. The performance of the database shall remain consistent as the number of entries grows.
- 3.3. The database shall allow an increase of more servers to distribute the workload as the user base grows.
- 3.4. The database shall be able to expand with new added features.
- 3.5. As the database grows the amount of memory allocated to the tables should grow with it.

4. Capability

- 4.1. The database shall support different data types.
- 4.2. The database shall query to find the needed information efficiently.

- 4.3. The database shall support multiple languages.
- 4.4. Applications should be able to integrate the database system.
- 4.5. The database shall work with many computer and server setups.

5. Environmental

- 5.1. The database shall run in a reliable space.
- 5.2. The database shall optimize its performance when it can.
- 5.3. The database shall run as a parallel database.
- 5.4. The data center locations shall be set up in areas with high connectivity outreach.
- 5.5. The database shall be available around the world.

6. Coding Standards

- 6.1. A coding style and format shall be used across the entire system.
- 6.2. Code shall be reusable when it can be to reduce redundancy.
- 6.3. Code will have comments to explain.
- 6.4. All errors will be documented and handled.
- 6.5. All code will be reviewed to see if it's up to coding standards.

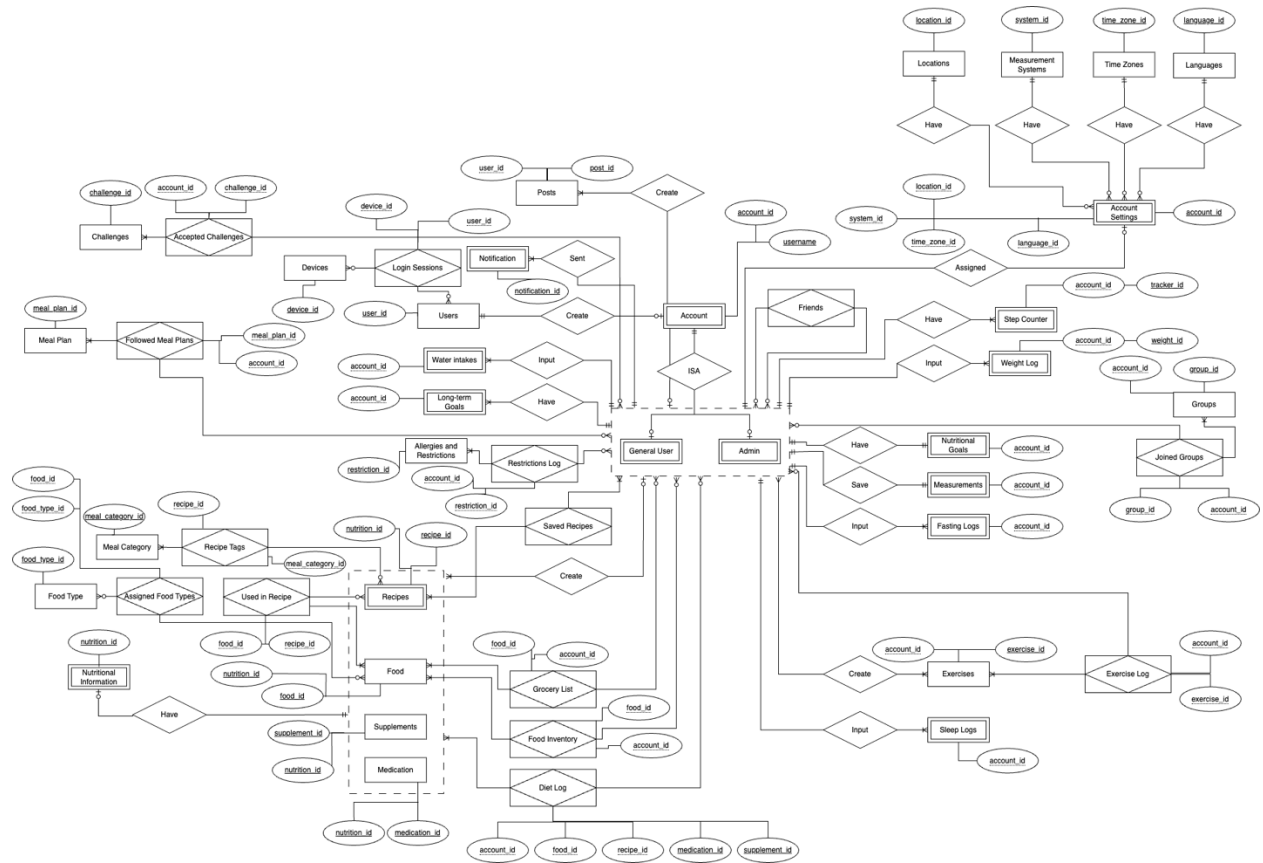
7. Media Storage

- 7.1. Different form of media shall be stored on the database.
- 7.2. Different file types of media shall be stored on the database.
- 7.3. Media stored shall be compressed to reduce size.
- 7.4. All media shall have a 200 MB limit.
- 7.5. All media shall have submission dates stored with.

8. Privacy

- 8.1. All data stored will be in compliance with the user.
- 8.2. If requested from the user, an admin shall delete their data.
- 8.3. No user can access another user's private information.
- 8.4. All data transferred to and from the database shall be encrypted.
- 8.5. Upon request of the user, their data can be sent to them.

Entity Relationship Diagram



Entity Description

1. Users (Strong)
 - * user_id: strong key, numeric
 - * first_name: alphanumeric
 - * last_name: alphanumeric
 - * name: composite, alphanumeric
 - * email: alphanumeric
 - * password: alphanumeric
2. Devices (Strong)
 - * device_id: strong key, numeric
 - * device_type: alphanumeric
 - * device_name: alphanumeric
3. Login Sessions (Weak)
 - * device_id: weak key, numeric
 - * user_id: weak key, numeric
 - * login_date: timestamp
 - * logout_date: timestamp
4. Account (Weak)
 - * account_id: strong key, numeric
 - * user_id: strong key, numeric
 - * username: alphanumeric
 - * admin: boolean
 - * dob: date
 - * age: derived, numeric
 - * date_joined: date
5. General User (Weak)
 - * account_id: strong key, numeric
 - * date_joined: date
 - * admin: boolean

6. Admin (Weak)
 - * account_id: strong key, numeric
 - * date_joined: date
 - * admin: boolean
7. Allergies and Restrictions (Strong)
 - * restriction_name: alphanumeric
 - * restriction_id: strong key, numeric
 - * restriction_type: alphanumeric
8. Restriction Log (Weak)
 - * severity_level: numeric
 - * restriction_id: weak key, numeric
 - * account_id: weak key, numeric
9. Time Zones (Strong)
 - * time_zone_name: alphabetic
 - * time_zone_id: strong key, numeric
 - * uct_offset: numeric
10. Locations (Strong)
 - * location_name: alphabetic
 - * location_id: strong key, numeric
 - * currency: alphanumeric
11. Measurement Systems (Strong)
 - * system_name: alphabetic
 - * system_id: strong key, numeric
 - * units: alphanumeric
12. Languages (Strong)
 - * language_id: strong key, numeric
 - * language_name: alphanumeric
 - * keyboard_name: alphanumeric
13. Account Settings (Weak)

- * language_id: weak key, numeric
- * system_id: weak key, numeric
- * location_id: weak key, numeric
- * time_zone_id: weak key, numeric
- * account_id: strong key, numeric

14. Measurements (Weak)

- * account_id: weak key, numeric
- * height: numeric
- * starting_weight: numeric
- * gender: alphabetic

15. Long-Term Goals (Weak)

- * account_id: weak key, numeric
- * title: alphanumeric
- * description: alphanumeric
- * date_assigned: date
- * complete_by_date: date
- * completed: boolean

16. Nutritional Goals (Weak)

- * account_id: weak key, numerical
- * calories: numeric
- * macro_nutrients: alphanumeric
- * micro_nutrients: alphanumeric

17. Weight Log (Weak)

- * account_id: key, numeric
- * weight: numerical
- * weight_log_date: date

18. Fasting Log (Weak)

- * account_id: key, numeric
- * time_length: numeric

- * fasting_date: timestamp
19. Water Intake (Weak)
- * account_id: weak key, numeric
 - * water_amount: numeric
 - * water_intake_date: timestamp
20. Step counter (Weak)
- * account_id: weak key, numeric
 - * step_amount: numeric
 - * step_counter_date: date
21. Sleep Log (Weak)
- * account_id: weak key, numeric
 - * sleep_amount: numeric
 - * sleep_date: date
22. Food (Strong)
- * name: alphanumeric
 - * brand: alphanumeric
 - * food_id: strong key, numeric
 - * nutrition_id: strong key, numeric
23. Food Type (Strong)
- * food_type_id: strong key, numeric
 - * food_type_name: alphanumeric
 - * food_type_description: alphanumeric
24. Assigned Food Types (Weak)
- * food_id: weak key, numeric
 - * food_type_id: weak key, numeric
 - * date_updated: date
25. Used in Recipe (Weak)
- * food_id: weak key, numeric
 - * recipe_id: weak key, numeric

- * date_updated: date

26. Grocery List (Weak)

- * food_id: weak key, numeric
- * account_id: weak key, numeric
- * priority_level: numeric
- * completed: boolean

27. Food Inventory (Weak)

- * food_id: weak key, numeric
- * account_id: weak key, numeric
- * expiration_date: date
- * gone: boolean

28. Medication (Strong)

- * name: alphanumeric
- * medication_id: strong key, numeric
- * instructions: alphanumeric
- * nutrition_id: strong key, numeric

29. Supplements (Strong)

- * name: alphanumeric
- * brand: alphanumeric
- * supplement_id: strong key, numeric
- * nutrition_id: strong key, numeric

30. Recipe (Weak)

- * name: alphanumeric
- * recipe_id: strong key, numeric
- * directions: alphanumeric
- * nutrition_id: strong key, numeric
- * difficulty: numeric

31. Diet Log (Weak)

- * account_id: weak key, numeric

- * food_id: weak key, numeric
- * recipe_id: weak key, numeric
- * medication_id: weak key, numeric
- * supplement_id: weak key, numeric
- * time_entered: timestamp
- * entity_type, alphabetic

32. Saved Recipes (Weak)

- * recipe_id: weak key, numeric
- * account_id: weak key, numeric
- * bookmark_name: alphanumeric

33. Meal Plans (Strong)

- * meal_plan_id: strong key, numeric
- * meal_plan_name: alphanumeric
- * meal_plan_description: alphanumeric

34. Followed Meal Plans (Weak)

- * account_id: weak key, numeric
- * meal_plan_id: weak key, numeric
- * date_subscribed: date

35. Meal Category (Strong)

- * meal_category_id: strong key, numeric
- * meal_category_name: alphanumeric
- * meal_category_description: alphanumeric

36. Recipe Tags (Weak)

- * meal_category_id: weak key, numeric
- * recipe_id: weak key, numeric
- * date_updated: date

37. Exercises (Strong)

- * exercise_id: strong key, numeric
- * exercise_name: alphanumeric

- * exercise_description: alphanumeric

38. Exercise Log (Weak)

- * account_id: weak key, numeric
- * exercise_id: weak key, numeric
- * date_entered: timestamp

39. Groups(Strong)

- * group_id: strong key, numeric
- * account_id: weak key, numeric
- * group_name: alphanumeric
- * group_description: alphanumeric
- * date_created: date

40. Joined Groups (Weak)

- * account_id: weak key, numeric
- * group_id: weak key, numeric
- * date_joined_group: date

41. Posts (Strong)

- * account_id: weak key, numeric
- * post_id: strong key, numeric
- * post_title: alphanumeric
- * post_description: alphanumeric
- * date_created: timestamp

42. Notification (Weak)

- * notification_id: strong key, numeric
- * account_id: weak key, numeric
- * notification_description: alphanumeric

43. Challenges (Strong)

- * challenge_id: strong key, numeric
- * challenge_name: alphanumeric
- * challenge_description: alphanumeric

- * challenge_length: numeric
- * challenge_type: alphanumeric

44. Accepted Challenges (Weak)

- * challenge_id: weak key, numeric
- * account_id: weak key, numeric
- * date_started: date
- * date_ended: date
- * completed: boolean

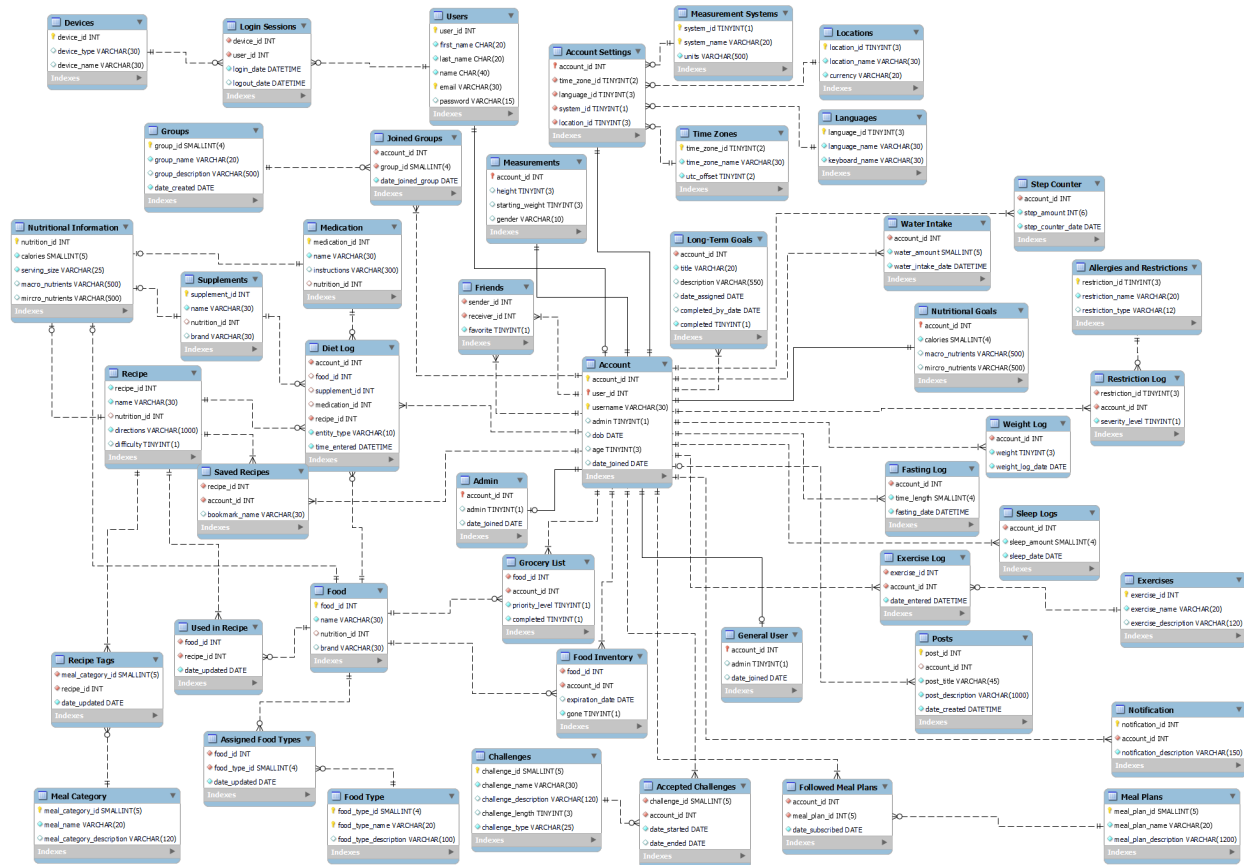
45. Nutritional Information (Weak)

- * nutrition_id: strong key, numeric
- * calories: numeric
- * serving_size: alphanumeric
- * macro_nutrients: composite, alphanumeric
- * micro_nutrients: composite, alphanumeric

46. Friends (Weak)

- * sender_id: weak key, numeric
- * receiver_id: weak key, numeric
- * favorite: boolean

Entity Establishment Relationship Diagram



Constraints Description

Table	FK	ON DELETE	ON UPDATE	Comment
Login Sessions	user_id	CASCADE	CASCADE	A log in session entry must have a user.
Login Sessions	device_id	CASCADE	CASCADE	A log in session entry must have a device.
Account	user_id	CASCADE	CASCADE	The user can have at most one account, so if the user is deleted the account must as well.
General User	account_id	CASCADE	CASCADE	A general user must have an account.
Admin	account_id	CASCADE	CASCADE	An admin must have an account.
Restriction Log	account_id	CASCADE	CASCADE	A restriction log entry must have an account.
Restriction Log	restriction_id	CASCADE	CASCADE	A restriction log entry must have a restriction.
Measurements	account_id	CASCADE	CASCADE	Measurements must have an account linked.
Account Settings	account_id	CASCADE	CASCADE	Account settings must have an account linked.
Account Settings	language_id	CASCADE	CASCADE	Account settings must have a language.
Account Settings	time_zone_id	CASCADE	CASCADE	Account settings must have a time zone.
Account Settings	system_id	CASCADE	CASCADE	Account settings must have a measurement system.
Account Settings	location_id	CASCADE	CASCADE	Account settings must have a location.
Long Term Goals	account_id	CASCADE	CASCADE	The long term goal must have only one account.
Nutritional Goals	account_id	CASCADE	CASCADE	The nutritional goal must have only one account.

Weight Log	account_id	CASCADE	CASCADE	The weight log must have only one account.
Fasting Log	account_id	CASCADE	CASCADE	The fasting log must have only one account.
Water Intake	account_id	CASCADE	CASCADE	The water intake must have only one account.
Step Counter	account_id	CASCADE	CASCADE	The step counter must have only one account.
Sleep Log	account_id	CASCADE	CASCADE	The sleep log must have only one account.
Assigned Food Types	food_id	CASCADE	CASCADE	The assigned food type must have only one food.
Assigned Food Types	food_type_id	CASCADE	CASCADE	The assigned food type must have only one food type.
Used In Recipe	food_id	CASCADE	CASCADE	The used in recipe entry must have one food.
Used In Recipe	recipe_id	CASCADE	CASCADE	The used in recipe entry must have only one recipe.
Grocery List	food_id	CASCADE	CASCADE	The grocery list entry must have one food.
Grocery List	account_id	CASCADE	CASCADE	The grocery list entry must have one account.
Food Inventory	account_id	CASCADE	CASCADE	The grocery list entry must have one account.
Food Inventory	food_id	CASCADE	CASCADE	The grocery list entry must have one food.
Food	nutrition_id	SET NULL	CASCADE	A food can have at most one nutritional information linked to it.
Medication	nutrition_id	SET NULL	CASCADE	A medication can have at most one nutritional information linked to it.

Supplements	nutrition_id	SET NULL	CASCADE	A supplement can have at most one nutritional information linked to it.
Recipe	nutrition_id	SET NULL	CASCADE	A recipe can have at most one nutritional information linked to it.
Diet Log	account_id	CASCADE	CASCADE	A diet log must have one account.
Diet Log	food_id	CASCADE	CASCADE	A diet log must have only one food, recipe, supplement, or medicine.
Diet Log	recipe_id	CASCADE	CASCADE	A diet log must have only one food, recipe, supplement, or medicine.
Diet Log	supplement_id	CASCADE	CASCADE	A diet log must have only one food, recipe, supplement, or medicine.
Diet Log	medicine_id	CASCADE	CASCADE	A diet log must have only one food, recipe, supplement, or medicine.
Saved Recipes	account_id	CASCADE	CASCADE	A saved recipe must have one account.
Saved Recipes	recipe_id	CASCADE	CASCADE	A saved recipe must have one recipe.
Followed Meal Plans	meal_plan_id	CASCADE	CASCADE	A followed meal plan entry must have one meal plan.
Followed Meal Plans	account_id	CASCADE	CASCADE	A followed meal plan entry must have one account.
Recipe Tags	recipe_id	CASCADE	CASCADE	A recipe tag entry must have one recipe.
Recipe Tags	meal_category_id	CASCADE	CASCADE	A recipe tag entry must have one meal category.
Exercise Log	exercise_id	CASCADE	CASCADE	An exercise log must have one exercise.
Exercise Log	account_id	CASCADE	CASCADE	An exercise log must have one account.

Joined Groups	account_id	CASCADE	CASCADE	A joined group entry must have one account.
Joined Groups	group_id	CASCADE	CASCADE	A joined group entry must have one group.
Posts	account_id	SET NULL	CASCADE	A post can have at most one account linked to it.
Notification	account_id	CASCADE	CASCADE	A notification must have one account linked to it.
Accepted Challenges	account_id	CASCADE	CASCADE	An accepted challenges entry must have one account linked to it.
Accepted Challenges	challenge_id	CASCADE	CASCADE	An accepted challenges entry must have one challenge linked to it.
Friends	sender_id	CASCADE	CASCADE	A friend entry must have one account linked to sender.
Friends	receiver_id	CASCADE	CASCADE	A friend entry must have one account linked to receiver.