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

Table of Contents

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. User: Strong
    1.1. A user shall create many accounts.
    1.2. A user shall login to many devices.
    1.3. A user shall have at least one role.
2. Account: Strong
    2.1. An account shall be created by at most one user.
    2.2. An account shall be a general user or an admin.
3. General User: Weak
    3.1. A general user shall save at least one set allergies and restrictions.
    3.2. A general user shall save at least one set of measurement.
    3.3. A general user shall have at least one long-term goal.
    3.4. A general user shall have only one set of nutritional goals per weight input.
    3.5. A general user shall have many weight logs.
    3.6. A general user shall have only one food log per day.
    3.7. A general user shall have only one day log per day.
    3.8. A general user shall input many fasting logs.
    3.9. A general user shall input many diary entries.
    3.10. A general user shall input many water intakes.
    3.11. A general user shall make many recipes.
    3.12. A general user shall input many foods in their food log.
    3.13. A general user shall input many recipes in their food log.
    3.14. A general user shall create many exercises.
    3.15. A general user shall input many exercises in their exercise log.
    3.16. A general user shall have many meal plans.
    3.17. A general user shall input many foods to their grocery list.
    3.18. A general user shall input many foods to their food inventory.
    3.19. A general user shall have many friends.
    3.20. A general user shall be in at most one family.

3.21. A general user shall create many posts.

3.22. A general user shall leave many comments on a post.

3.23. A general user shall make many reports.

3.24. A general user shall have only one step counter.

3.25. A general user shall have many sleep logs per day.

4. Admin: Weak

4.1. An admin shall do everything a general user can do.

4.2. An admin shall delete posts.

4.3. An admin shall delete comments.

4.4. An admin shall delete accounts.

4.5. An admin shall delete recipes.

4.6. An admin shall delete foods.

4.7. An admin shall access reports.

5. Allergies and Restrictions: Strong

5.1. A list of allergies and restrictions shall be assigned to only one account.

6. Measurements: Strong

6.1. A set of measurements shall be assigned to only one account.

7. Long-Term Goals: Strong

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

8. Nutritional Goals: Strong

8.1. A nutritional goal shall be assigned to only one account.

9. Weight Log: Strong

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

10. Sleep Log: Strong

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

11. Step Counter: Strong

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

12. Food Log: Weak

12.1. A food log shall be assigned to one account.

12.2. A food log shall hold many foods.

12.3. A food log shall take in many recipes.

13. Day Log: Weak

13.1. A day log shall record the food log of the day for each account.

13.2. A day log shall record the water intake of the day for each account.

13.3. A day log shall record the nutritional goals of the day for each account.

13.4. A day log shall record the step counter of the day for each account.

13.5. A day log shall record the sleep log of the day for each account.

13.6. A day log shall be assigned to one account.

14. Fasting Log: Strong

14.1. A fasting log shall be input by one account.

15. Diary Entry: Strong

15.1. A diary entry shall be input by one account.

16. Food: Strong

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

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

16.3. A food shall be used in many recipes.

16.4. A food shall be used in many food logs.

16.5. A food shall be in many grocery lists.

16.6. A food shall be in many food inventories.

17. Water Intake: Strong

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

18. Nutritional Information: Weak

18.1. A nutritional information shall be assigned to one food or recipe.

19. Recipes: Weak

19.1. A recipe shall have at least one food.

19.2. A recipe shall have at most one nutritional information.

19.3. A recipe shall be stored in many saved recipes.

19.4. A recipe shall be used in many meal plans.

19.5. A recipe shall be used in many food logs.

19.6. A recipe shall be stored in many food inventories.

19.7. A recipe shall be created by at most one account.

20. Saved Recipes: Weak

20.1. A saved recipe shall have only one recipe saved by only one account.

21. Exercises: Strong

21.1. An exercise shall be created by at most one account.

21.2. An exercise shall be inputted into an exercise log by one account.

22. Exercise Log: Weak

22.1. An exercise log shall have many exercises.

23. Groups: Strong

23.1. A group shall have many accounts.

24. Meal Plans: Weak

24.1. A meal plan shall have many recipes.

24.2. A meal plan shall be assigned to many accounts.

24.3. A meal plan shall hold many recipes.

25. Exercise Plans: Weak

25.1. An exercise plan shall be assigned to many accounts.

25.2. An exercise plan shall hold multiple exercises.

26. Grocery Lists: Weak

26.1. A grocery list shall be assigned to one account or family.

26.2. A grocery list shall hold many foods.

27. Food Inventory: Weak

27.1. A food inventory shall be assigned to one account or family.

27.2. A food inventory shall hold many foods.

27.3. A food inventory shall hold many recipes.

28. Friends: Weak

28.1. A friend shall link one account to another account.

29. Family: Weak

29.1. A family shall hold many accounts.

29.2. A family shall have only one food inventory.

29.3. A family shall have only one grocery list.

30. Posts: Strong

30.1. A post shall be assigned to at least one account.

30.2. A post shall have many comments.

30.3. A post shall have many reports.

31. Comments: Weak

31.1. A comment shall be assigned to one post by at least one account.

31.2. A comment shall be on only one post.

31.3. A comment shall have many comments.

31.4. A comment shall have many reports.

32. Reports: Weak

32.1. A report shall come from one account.

32.2. A report shall be on one comment or post.

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, their data can be deleted.

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.