

Team 6 - Green Prep

Jules Garrett, Collyn Gatton, Katie Hrenchir, Austin Juhl, Menita Vedantam

Project Synopsis

Meal preparation web application that determines the optimal serving sizes of user selected foods in order to meet nutrient goals for a meal. Our tool will make meal preparation quick, simple, and maintainable.

Project Description

Introduction

The three main nutrients in all foods are called macronutrients and include protein, carbohydrates, and fat. To maintain a healthy diet, you want to have a meal plan based on your specific macronutrient goals. Your macronutrient goals can be calculated using your gender, age, height, weight, activity level and your fitness goal for your diet whether it be losing fat, maintaining weight, or gaining muscle.

Problem:

Maintaining a healthy and balanced diet is difficult for many reasons. Most people are unaware of the nutritional makeup of the foods they eat everyday. Without knowing the nutritional value of the food you eat it is impossible to determine your daily nutrient intake and equally as difficult to determine what your body actually needs in a day. For those who are aware of the nutritional makeup of the foods they eat, determining the appropriate serving sizes depending on your personal needs is challenging and time consuming. Also, a byproduct of the difficulty related to meal preparation is the absence of variety and the miscalculation of quantity. Both of which can lead to last minute take-out or food waste. If eating healthy is tedious, unenjoyable, and wasteful you're not putting yourself in a position to achieve your nutritional goals.

Current Products:

Currently, there are nutrient apps that will tell you whether or not you met your goals after journaling what you ate at the end of the day. This reactive approach to dieting does not give you a plan to eat well rather it informs you whether or not you ate poorly.

Objective:

The main function of our nutrition app will be to optimize meals that meet the user's nutrient goals. This will be accomplished by creating a system of equations that corresponds to the nutrient break down of each food and includes lower and upper bounds. Once the system is defined we will use the Simplex Method to generate the optimal meal.

Secondary functions:

- Macronutrient calculator.
- Produce accurate grocery lists based on generated meal plans.
- Make food recommendations to create more optimal meals and utilize leftover food.
- Restrict food bank for dietary restrictions.

Project Milestones

- First Semester:
 - Choose tech stack - September 21, 2018
 - Front-end design workflows - October 2018
 - Back-end design workflows - October 2018
 - Define Database structure - October 2018
 - Wireframes for UI design - November 2018
- Second Semester:
 - Load database - February 2019
 - Front-end implementation - March 2019
 - Back-end implementation - March 2019
 - User Testing - April 2019
 - Revisions and bug-fixes - April 2019
 - Deployment - Late April 2019

Project Budget

- Domain Name: \$12 from Google Domains
- Webpage Hosting: free plan with Heroku
- There will be no hardware, software, and/or computing resource costs
- There will be no special training costs
- There will be no vendor costs

Work Plan

- Austin: Backend/Django, Testing, Database
- Coltyn: Frontend/React, Testing, Security
- Jules: Backend/Django, Testing, Database
- Katie: Front-end/React, Backend/Django, Testing
- Menita: Backend/Django, Testing, Team Status Reporter

Preliminary Project Design

User Interactions

When the user first uses the application, they will be prompted to create an account. While creating an account the user will create a username and password as well as provide their name, weight, height, age, current activity level, their physical goal while using the application (e.x. lose weight). Users will also have the option to choose from common diets like keto, Atkins, or protein-heavy diets. Based on this information, the application will give the user a recommended macro amount. The user can also decide to input their own Macros instead of going based off the suggestion. The user will be able to change these values later, allowing for their Macro plan to adapt based on the user's progress or if the user changes their goal.

Once the user has established their daily nutrient goals they will determine how many meals they would like to eat in a single day up to 10 meals. Green prep will then evenly divide the daily nutrients amongst the determined number of meals. The user may continue with an even distribution of nutrients or they can choose to redistribute how they see fit. For example, a user may want to concentrate more carbs around their workout and less carbs before bed. Next, the user will select the foods they wish to eat for each meal by searching the food database. Curated meals will be available for users to choose from as well. The user also has the option to put upper bounds on the foods they select. This can be helpful if the user has a limited quantity of food, is under financial constraints, or if they simply wish to create a meal with more variety. Once the user has selected the foods they wish to eat for each meal, Green Prep will calculate the optimal serving size of each food in order to achieve the established nutrient goals. Additionally, Green Prep will report an optimization score for the meal using a scale from 1 to 100 - this score is based on how close the nutrient composition of the meal is to the set goals. If a meal rating is below the user established quality minimum, Green Prep may make a food suggestion to improve the meal.

Green Prep will do its best to suggest optimal meals, but users will be able to override any decisions its algorithm makes. If a user exceeds their daily limits in any of the three macro categories, they will be altered. Once the user is happy with the meal plan they've created they can save it and add it to their calendar.

After the user has compiled their meal plan, a grocery list will be generated based upon the meals they have selected. Users can select the time frame for which they want their grocery list to be generated - for a single day, for a handful of days, or for multiple weeks, assuming they have completed meal plans for those day. Foods that are used across multiple meals/recipes will be consolidated in the shopping list. For example, both chicken salad and chicken kebabs involve chicken, so the individual measurements of chicken those recipes call for would be added together. This also allows users to gauge the amount of an ingredient their plan requires

compared to common packaging sizes and adjust their purchases accordingly. This will minimize waste from over-purchasing ingredients.

Explanation of Calculations - Simplex Method

Green Prep calculates the optimal serving size of each food using the Simplex Method. Generally, how the Simplex Method works is by starting at the origin of the polytope defined by the constraints on the system. In our case, the nutrient composition of each food and the nutrient goals serve as both the constraints and objective equations. Next, we move along the edges of the polytope while staying in the feasible region. As we move along the polytope edges we observe the neighboring corners and move to the one that best improves the solution. Once we find the most optimal corner of the feasible region we have our solution. It's important to note that any Linear Programming Problem must have an optimal solution that corresponds to a corner, but there may be alternative optimal solutions.

Programmatically we represent moving about the polytope using a tableau and Gaussian Elimination.

To create the tableau we start by turning the linear inequalities constraints into equalities using slack variables as well as assigning the objective equations to their own variable. For example...

Pizza:	8.63p,	30.22c,	10.79f
Chicken:	18.75p,	0c,	3.57f
Slack Variables:			
Goal:	50p,	40c,	20f

8.63	18.75	1	0	0	0	0	0	0	50
30.22	0	0	1	0	0	0	0	0	40
10.79	3.57	0	0	1	0	0	0	0	20
-8.63	-18.75	0	0	0	1	0	0	0	0
-30.22	0	0	0	0	0	1	0	0	0
-10.79	-3.57	0	0	0	0	0	1	0	0

Figure 1 - Initial Tableau

Next, find the pivot column by selecting the most negative value in the objective functions located at the bottom half of the tableau. In our example we would select column 1 with a minimum value of -30.22.

Then, we find the pivot row by traversing the constraint values in the pivot column and if the value is positive we find the quotient of last column value and pivot column value. The minimum quotient gives us the pivot. In our example we find minimum quotient 1.3236 in the second row. Then we divide the pivot row by the pivot value and perform Gaussian Elimination to make every other value in the pivot column zero. We repeat these steps until all non slack values in the objective functions are positive.

The final tableau will have the **optimal serving size of each food** in the top right corner and the **nutrient composition of the optimized meal** in the bottom right corner.

0	0	1	1.5897	-5.2521	0	0	0	8.5452
1	0	0	0.0331	0	0	0	0	1.3236
0	1	0	-0.1	0.2801	0	0	0	1.6017
0	0	0	-1.5897	5.2521	1	0	0	41.4548
0	0	0	1	0	0	1	0	40
0	0	0	0	1	0	0	1	20

Figure 2 - Final Tableau

Database

The database for our application will consist of eleven tables utilizing a SQL server. One of the main tables is tblUser. In this table general information that the user inputs when they first create an account will be stored. The user's activity level and physical goal (e.x. lose weight) will be selected by the user based off the options provided in tblActivityLevel and tblPhysicalGoal.

The user will also access tblMeal. The meal table will contain the meals that the user has created. tblMeal is connected to tblMealFood where food is associated with the meal it is contained in.

TblMealFood will access two tables to gather all the nutritional information needed for our algorithm. The first table it will access is the tblUnit. TblUnit will contain all the different measurements of food. The second table tblMealFood will access is tblFood. TblFood will be a table containing all the food the user can choose to add to their meal, along with all the nutritional information for the food. TblFood will be created based off the USDA's food database that contains an extensive list of different food and all the nutritional information related to the food.

The user will also have access to tblGrocery. TblGrocery is where any grocery lists the user has made will be stored. TblGrocery will have access to the tblGroceryMeal. This table

associates meals with the grocery lists it is contained in. Using the meal id we can compile the foods stored in tblMealFood to create an accurate grocery list.

TblUser also accesses tblNutrientGoal. TblNutrientGoal will contain the three nutrients the user wants to optimize when building their meals as well as the goal value of that nutrient and the unit of measurement for the nutrient. TblNutrientGoal will access tblNutrient, where the names of all the nutrients that the user can optimize are stored.

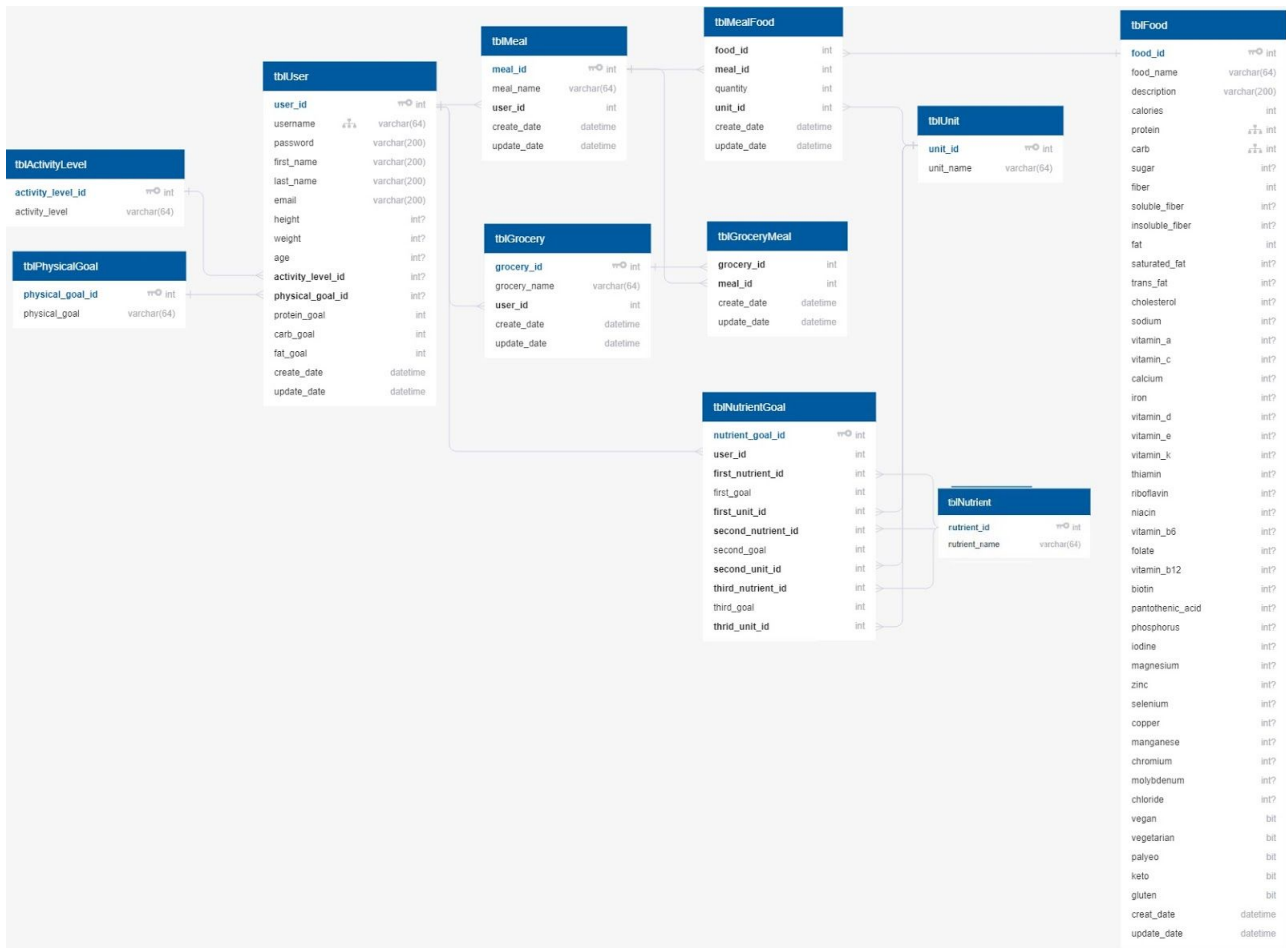


Figure 3 - Database structure

Front-end Design

The front-end of this project will be implemented using the end result of the proj Javascript and React. React is a Javascript library for building web-based user interfaces using *components*, which are small chunks of code that are combined to form a cohesive interface. React works in combination with traditional web-programming technologies like HTML and CSS to create modern, extensible web projects.

Each component takes in parameters, called *props* and returns a *view* to display on the screen using *renders*. Our project will consist of four main parts, in which a user can: amend their dietary preferences or profile information, create or edit their meal plans, view completed meal plans, or generate their shopping list. To access these views, a user must first complete the initial profile information - this short quiz will be displayed to users immediately after creating an account. The diagram below shows the flow of views.

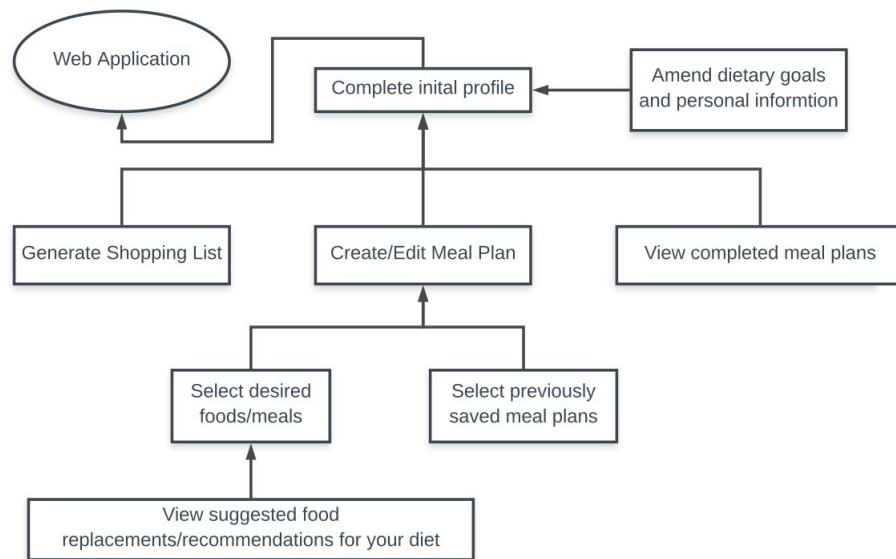


Figure 4 - Flow chart of user views

These four views make up the main functionality of the program. Within each view, users will be able to interact with their meal plan, grocery list, or dietary goals.

Ethical and Intellectual Property Issues

In order for our project to work well we will have to request some rather personal information. For example, we will be collecting names, emails, ages, height, weight, and a few other things so users can create an account and have a more personalized experience. With this information comes a lot of ethical responsibility especially with so many account breaches happening recently. We will be spending some time making sure this information is safe and well kept. Another ethical issue is that we are not nutritionists and our tool might not work for everyone. To help prevent people from assuming this we will be creating a disclaimer that goes with our web application.

We do not expect to have many intellectual property issues within our project. This is because we plan on writing all of the code from scratch and using free open source software to help us with our project. To help protect our code and ideas from infringement we as a team

decided that we are going look into registering a patent for our software. The patent will be a difficult to acquire because the idea needs to be unique, but our team believes that this application is a new and unique way to look at nutrition calculation. We will be looking into the process of applying for a patent.

Change Log

Timeline

We altered our timeline to accommodate for our course schedules and other commitments. We also lengthened our design process to allow for more thorough designs.

Macronutrient/General Nutrient Tracking

We decided to have our application only optimize the three macronutrients, because these are the three main nutrients in all foods. This also keeps the user from trying to optimize a nutrient like sugar to a user inputted amount, which is typically minimized.

Secondary Functions

We removed dynamic meal adjustment and sharing of meal plans from our secondary goals because we did not believe that those features fit within the scope of the this project given the time constraints.

Warning Flags

We decided to add a flag if the user goes over daily recommended servings of other nutrients. We did this to give the user more information while still allowing the user the freedom to choose the food they want.

Gantt Chart

<https://docs.google.com/spreadsheets/d/1syD-vRhOyLZCIfMPYxuHkL9TUcScphRiTt79KTHbeI/edit?usp=sharing>