

Diet Prescription Calculator for Precision Ketogenic Therapy

Dylan Booth

CIS4914, Senior Project

Department of CISE

University of Florida

Advisor: Dr. Peggy Borum, *email:* prb@ufl.edu

Department of Food Science and Human Nutrition

University of Florida, Gainesville, FL 32611

Abstract

For over 50 years, Precision Ketogenic Therapy (PKT) has been used by healthcare providers to reduce seizures in patients of all ages. This process uses detailed information about a patient's diet to precisely monitor their intake of fat, protein, and carbohydrates, as well as calories and other nutrients. The result is not only a reduction in seizures but also in improvement in lifestyle for these patients. Since PKT has been used for many years, its implementation has evolved with technology. What was originally maintained through pen-and-paper notes is now being stored across many Microsoft Excel spreadsheets. This has made it easier to share new methods of PKT between healthcare providers. However, the current implementation can be improved upon.

Our group's project seeks to solve these issues by consolidating the many aspects of PKT into a web application that will facilitate use. This application will be coded in Python with a Flask framework and an HTML frontend. The backend will consist of a PostgreSQL database which will store the necessary patient data. Our individual projects will be stored in docker containers and uploaded to Resvault to be used by PKT clinical members.

1. Introduction.

The current methods used for Precision Ketogenic Therapy are slow to load, difficult to learn, and easily broken. These problems are caused by large spreadsheets, overflow of data, and formulas that can be accidentally deleted. The goal of our application is to solve these problems by storing data in a relational database and creating a frontend site that is intuitive and robust. Our group project will consist of a homepage that will link to our individual projects, which are different elements of PKT. The team working on the data entry part of the application consists of Devin Blem's Foodomics Database, Gregory Scott Wilkinson's Metabolite Calculator, Tyler Mueller's Intake Calculator, Dylan Booth (the author)'s Diet Prescription Calculator, Fareed Khamitov's Meal Equivalents Calculator, and Adam Weizman's Cookbook application. The rest of our team is working on data display and visualization. It consists of Berlin Sankar's Healthcare Data Visualization and Paige Applegate's Patient Data Visualization. Our team may be large, but frequent meetings will help to ensure group members are communicating and working on track.

The Diet Prescription Calculator will allow the user to enter clinical data for the patient and prescribe a new diet while displaying the data from the patient’s previous prescription. The calculator will also provide references and calculations to help the user when prescribing a new diet.

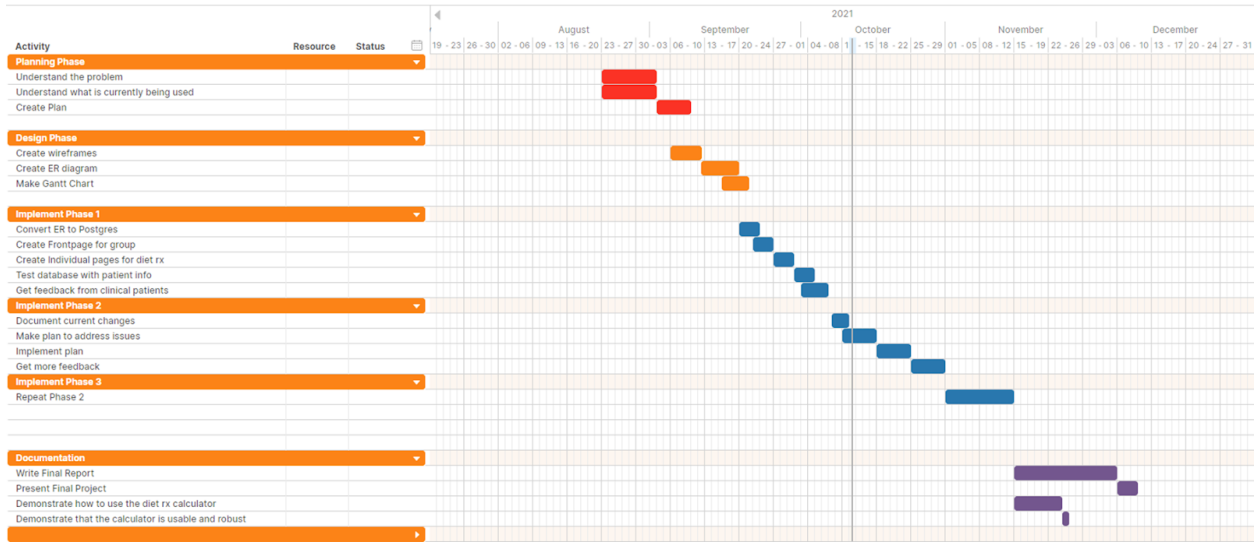


Figure 1. The Gantt Chart for the Diet Prescription Calculator

The schedule for the development of the Diet Prescription Calculator is displayed in Figure 1. This Gantt Chart is separated into four phases: planning, designing, implementing, and documentation. The planning phase will be used for research. The design phase consists of creating current diagrams. The implementation phase will entail coding the application and sending it to clinical students for feedback. Finally, the documentation phase will involve concluding the project through written reports and submissions. Maintaining this schedule will be ideal, but interruptions could occur when testing in a virtual environment, implementing a new Postgres database, pulling from the database of another group member, etc. Any problems of these types should be discussed with other group members to find solutions quickly.

2. Problem Domain.

The current methods being used for PKT, including those for diet prescription, are outdated and have potential user flaws. All the calculators are contained in Microsoft Excel spreadsheets, but the files are spread out through the lab’s Microsoft Teams groups, making it difficult to update the necessary file or upgrade the desired template. Many of the calculators pull data from other

spreadsheets, such as columns from the Foodomics database, which makes them slow to load. This is not only frustrating for the user but also problematic if patient data needs to be retrieved quickly. There have also been multiple data science members contributing to these spreadsheets at different times, causing extraneous tables to remain unfinished and undocumented within the calculators. This problem has been mitigated by detailed explanations in the Data Dictionary tabs, though some changes remain untracked. Finally, many user errors have occurred by accidentally deleting an Excel formula, which can also delete important patient data or break another part of the calculator. Our project will solve these issues by using a relational database that will load faster, contain less extraneous information, and check for errors in user input.

3. Previous Work (Literature Search).

As previously stated, the basis of this project is to improve the system that is already being used for prescribing PKT diets. The data that is recorded in the new diet prescription calculator was also recorded in the previous diet prescription calculator [1], which was maintained in Microsoft Excel. Learning about previous methods of diet prescription was useful in determining what data the users should record. Notably, the reference tables were also transferred from the previous calculator. However, diet prescription is not the only part of PKT. Researching an overview of PKT and its implementations demonstrated the needs that needed to be met by the diet prescription calculator [2]. This dissertation also outlined how the different part of PKT should interact at a high level.

An important element of this project was the database to not only store and retrieve rows for the diet prescription calculator, but to also make the data available for other group members to retrieve in their projects. As a group, we chose to use PostgreSQL to have a consistent method of retrieving data from the database tables. The documentation for PostgreSQL [3] was used to understand the syntax and structure necessary to implement our databases. Similarly, the necessary HTML and JavaScript syntax used to create the web pages was assisted by W3 School Tutorials [4].

Necessary formulas for body-mass index were researched from the CDC [5]. IEEE Standards [6] were also important for maintaining a consistent format. Miscellaneous errors and unknown syntax were researched in Stack Overflow forums, such as finding the maximum value in a database column [7].

4. Technical Approach (Solution).

As a group, our focus was designing an application that would make the many parts of PKT more accessible by maintaining a consistent design and flow.

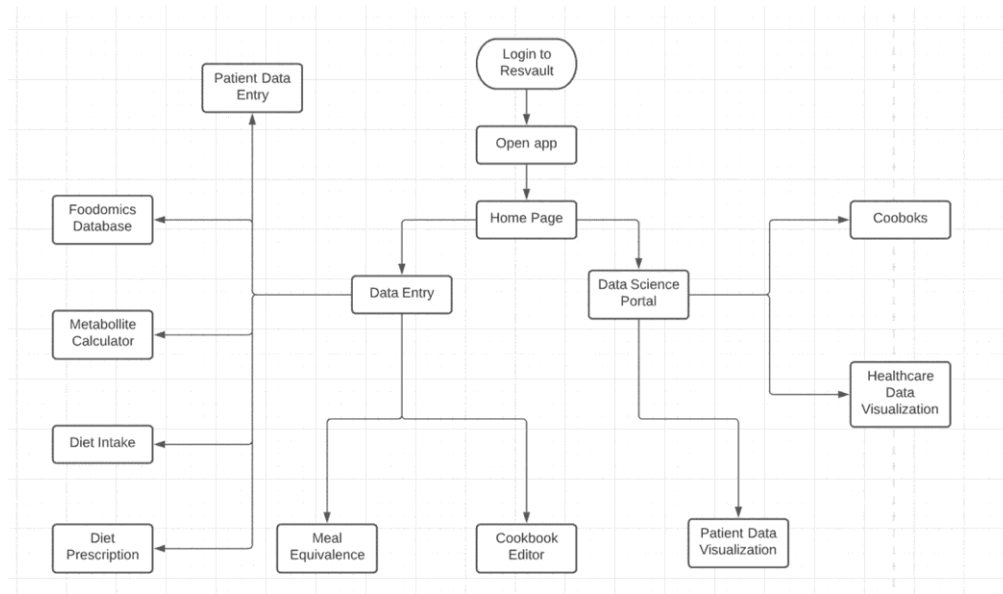


Figure 2. The group flowchart for the home page

Figure 2 shows the flowchart for the group's homepage. After logging in and opening the app, the user is presented with a choice between Data Entry and Data Science. Selecting the Data Entry option will allow the user to choose from the projects that require entering data. Similarly, selecting the Data Science option will show the projects that display data. This design is efficient because a user may need to enter data into many parts sequentially. For example, the user may enter a new patient into the database through the Patient Data Entry section, then update the past week of diet intakes, add new diet prescriptions, and finally assemble a new page of recipes. These options are all easily accessible to one another through the application.

The Diet Prescription Calculator is divided into separate different pages. The first page is called Clinical Data Entry, which is where clinical members will update patient data. The other page is called New Diet Prescription, which allows the user to input new values, such as calories and grams of protein per day, while comparing these values to the previously prescribed. This page also pulls from reference tables which compare the patient's anthropometrics to the ideal values for their sex and age. A more detailed flowchart is displayed in Figure 3. It expands the Diet

Prescription element of Figure 2. This flowchart details the interactions that a user may have when entering data into the calculator. The key for this diagram is displayed in the upper-left corner, which simplifies the flowchart into actions made by the user and responses from the application.

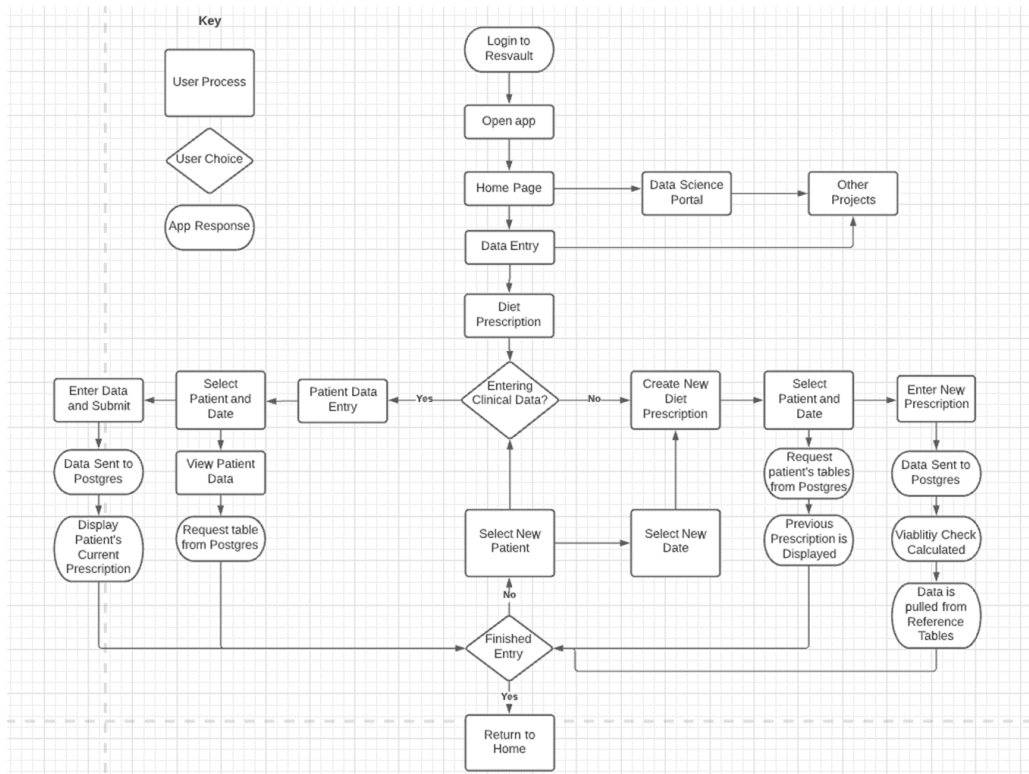


Figure 3. The Diet Prescription Flowchart.

To store the clinical data that will be used in the application, our group agreed to design a consistent PostgreSQL database to be accessed across different projects. The columns that would make up the tables for this database were first detailed in an Entity Relationship Diagram (ERD). The ERD for the Diet Prescription Calculator is displayed in Figure 4. The Diet Rx table contains the data stored in a new diet prescription, which consists of data entered by the user, calculations based on this data, and explanations to record the change. Similarly, the Diet Entry table stores data from the Clinical Data Entry page. The reference tables form a static database that is queried when creating a new diet prescription to provide context for the user regarding the health of the patient. The Diet Rx and Diet Entry tables have a many-to-many relationship, meaning that a patient can have multiple clinical entries per diet prescription, or multiple diet prescriptions per clinical entry.

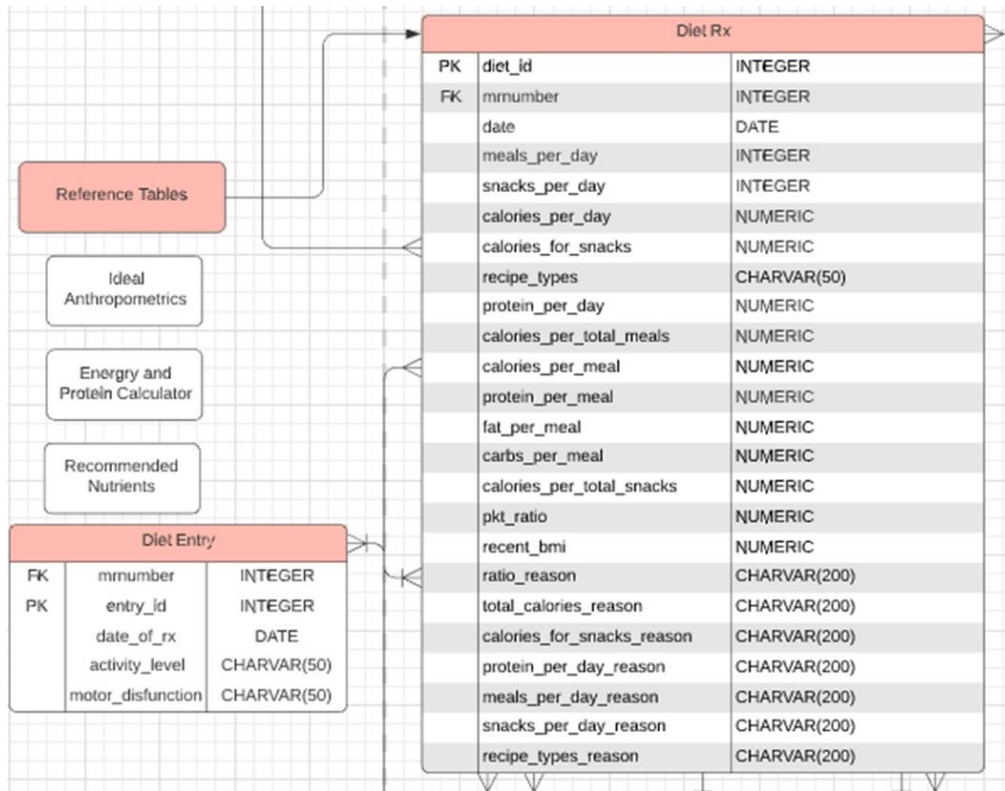


Figure 4. The Entity Relationship Diagram for the Diet Prescription Calculator, separated into different tables.

As a result of having a large app development team, the group’s ERD is too large to fit on a single page and will not be provided here. Therefore, an explanation of the relationships between the Diet Rx tables and the other projects will be provided instead. The Diet Rx table has many-to-many relationships with the Diet Intake and Meal Equivalent tables. This is because a patient can have many intakes recorded or equivalent meals created per diet prescription, and vice versa.

4.5. Results

The determined solution for this problem was a web app that would store the user’s input in the Postgres database and display previous values. An example from the New Diet Prescription page is displayed in Figure 5. The user is routed to this page after inputting the medical record number (MRN) of a patient and selecting the *New Diet Prescription* option in the menu bar. For convenience, a test patient has been displayed with fake data.

Prescribe a New Diet

Current Patient: Booth, Dylan

MRN: 1

mm/dd/yyyy

Components of Diet Prescription	Proposed Diet Prescription	Current Diet Prescription	Reason for Change	Data Dictionary Link
PKT Ratio	Pkt_ratio	4.00	Pkt_ratio explanation	?
Total Calories per Day	Calories per day	3200.00	Calories per day explanation	?
Calories for Snacks per Day	Calories for snacks	1000.00	Calories for snacks explanat	?
Grams of Protein per Day	Protein per day	63.00	Protein per day explanation	?
Number of Meals per Day	Meals per day	3	Meals per day explanation	?
Number of Snacks per Day	snacks per day	2	Snacks per day Explanation	?
Types of Recipes	recipe types	r	Recipe Types	?

Is the Diet Prescription Viable?	Proposed Value	Currently Prescribed Value	Data Dictionary Link
Calories per Total Meals per Day		2200.00	?
Calories per Meal per Day		733.33	?
Grams of Protein per Meal per Day		14.44	?
Grams of Fat per Meal per Day		73.33	?
Grams of Carbohydrate per Meal per Day		3.90	?
Calories per Snack per Day		500.00	?
Grams of Protein per Snack per Day		9.84	?
Grams of Fat per Snack per Day		50.00	?
Grams of Carbohydrate per Snack per Day		2.66	?

Save

Figure 5. An excerpt from the New Diet Prescription page

The menu bar at the top of the page allows the user to navigate through the Diet Prescription Calculator or return to the app’s home page. The name and MRN of the selected patient are always displayed at the top of the page so the user always knows which patient is displayed. A table structure is used to organize the data. This format also resembles that of the Excel Diet Prescription Calculator so it can be familiar to the clinical members who will be using it. The tables are separated so that the upper table is used to input data and the lower tables are used to display calculated data. The goal was to present an experienced user with enough data to make educated decisions for the prescriptions, but also to not overwhelm a new user who is learning to use multiple different calculators.

After testing, it was determined that a user may need to examine past diet prescriptions for a patient beyond the most recent prescription. An additional page was created in response, the Rx Records page, which displays all the patient’s past diet prescriptions chronologically. This solution will ensure that a user can find a previous prescription, without interrupting the rest of the application.

5. Technical Challenges

As fitting of a project with multiple interacting parts, several technical challenges were encountered before the solution was complete. Most notably was a fatal error involving writing

to a new table in the existing database from the application hosted in a virtual environment. This problem was particularly confusing, as attempting to run the application on a local machine created a different error. However, this problem was solved by restarting, uninstalling, and reinstalling PostgreSQL on the local machine. The original problem may have been caused by a missing dependency in the original Postgres download. Thorough development and testing began once this problem was solved.

Another challenge originated from importing large reference tables from inconsistent Excel formats into the Postgres database. Notably, these tables consisted of 500 rows and multiple columns, eliminating the option of manual retyping. The solution involved moving the data to open Excel columns separated by columns of commas. Copying the newly formed table allowed the Python file to write the data to the database when the user presses the *update* button.

6. Conclusion

The Diet Prescription Calculator is a component of the PKT application that stores and calculates data needed for prescribing a PKT diet. Developing this app required learning and implementing Python, Flask, HTML, and Postgres. Discussions with clinical members were insightful in learning what the user needed from this app. Frequent meetings with the app development team required concise communication to explain problems that arose and create group diagrams. After this was accomplished, the Diet Prescription Calculator could be used by clinical members.

6.5. Future Work

While the Diet Prescription Calculator is functional, some of its methods could be improved upon. The current method of displaying current or previous data displays the most recently data that was entered. If multiple diet prescriptions are entered at once, this may not display the data that the user anticipates. Customization options for data display could be helpful as a feature to be added.

Currently, not all the nutrients that a patient is taking in are being tracked. After the other group projects are also finished, an updated intake summary can be implemented. This intake summary could display the vitamins as well as the macronutrients and elements that a patient is taking in from their current diet. This vitamin data will also be necessary in prescribing supplements to

patients who are deficient in certain vitamins. An algorithm could be created to optimize a patient's nutrient intake by recommending supplements for deficient vitamins without exceeding tolerable limits. This feature would reduce unnecessary calculations and guessing required by the user.

7. Standards and Constraints

Programming was performed in Python version 3.9.9 and compressed with Docker version 20.10.8, adhering to IEEE standards. Modifications were tested on a local environment with an 8-core processor and 16 GB of RAM.

8. Acknowledgements

The author would like to thank the group's advisor, Dr. Peggy Borum, who not only facilitated communication, but also provided motivation and made clear the usefulness of this project. Without her guidance and expertise in the field of PKT, this project could not have been completed. Additional thanks to members of the clinical team who answered questions and provided a user's perspective as well as encouraging support: Kaitlyn Taylor, Harleen Kahlon, Samantha Waterman, and Bianca Barroso. Our group would also like to show gratitude towards the recently graduated students from the Borum Lab, Javier Moreno Flores and Octavio Ochoa, for setting up tutorials and explaining the previous application and calculators. Final thanks to the rest of the app development team for their contributions to the group's diagrams and technical recommendations. Specifically, thanks to Berlin Sankar for handling the pushes to Resvault and assisting with troubleshooting.

9. References

- [1] David Fussel, Simon Frank. 2018. *Diet_Prescription_template*. Microsoft Excel.
- [2] Allen, Hannah J. *Precision ketogenic therapy for seizures: Metabolic basis for change in nutrients changing phenotype*. University of Florida. ProQuest Dissertations Publishing, 2016.
- [3] "PostgreSQL 8.4.22 Documentation." *PostgreSQL Documentation*, 24 July 2014, <https://www.postgresql.org/docs/8.4/index.html>.

- [4] *Onload Event*, https://www.w3schools.com/jsref/event_onload.asp.
- [5] “CDC - Calculating BMI Using the Metric System - BMI for Age Training Course - DNPAO.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 9 May 2014, https://www.cdc.gov/nccdphp/dnpao/growthcharts/training/bmiage/page5_1.html.
- [6] “Using a Coding Standard to Improve Program Quality.” *IEEE Xplore*, <https://ieeexplore.ieee.org/document/990004>.
- [7] John and ford. “SQLALCHEMY 0.5.8 Max Function.” *Stack Overflow*, 1 Nov. 1958, <https://stackoverflow.com/questions/3750118/sqlalchemy-0-5-8-max-function>.

10. Appendix A – Technology Transfer Plan

After development has completed, each team member will push their respective projects to the group’s Github repository and merge the branches onto the main branch. From there, the group’s Resvault expert will push the updated application to Resvault’s remote machine. As new clinical and data science members join the Borum Lab, they will receive an access key to log in to Resvault and open the application. Clinical students will begin interacting with the application and other data science members will add to or upgrade it.

11. Biography

Dylan Booth was born in Riverview, Florida, and moved to Gainesville after completing high school to pursue an engineering degree from the University of Florida. He started on the path of Computer Engineering but decided to switch to a Computer Science degree supported by his minor in Business Administrations. He has worked as the camp intern at the Retreat at Silver Springs in Ocala, Florida, and will continue his involvement and leadership there after his graduation. He will also be applying for jobs in his field in the Gainesville and Tampa areas. Dylan’s goal is to create programs that will make a positive impact on peoples’ lives.