

Cookbook Creation and Organization for PKT Lab

Adam Weizman

CIS4914, Senior Project
Department of CISE
University of Florida

Advisor: Peggy R. Borum, Ph.D. *email:* prb@ufl.edu
Department of Food Science and Human Nutrition
572 Newell Drive
University of Florida, Gainesville, FL 32611

Date of Talk: 29 Nov 2021

Abstract

Dr. Peggy Borum and her team of lab partners have been working on a specific kind of ketogenic therapy since 1995¹, used to treat seizures in patients. The lab is based on research suggesting how ketogenic diets can be used to treat epilepsy and seizures, with Dr. Borum and her team publishing papers on the lab since 2006¹, with improvements being made to the system year after year². This year, their planned improvements were to move the infrastructure from a fragile collection of Excel spreadsheets to a database hosted on ResVault, which is currently being migrated to UF's HiPerGator 3.0 supercomputer. A collection of seniors including myself working on separate but related senior projects are working on making this idea a reality, using a PostgreSQL database to store the data previously being held in excel spreadsheets and a web-app frontend managed with the Flask micro-framework to allow lab partners to interact with the database.

The senior project described here describes how cookbooks for patients containing recipes that fit their dietary needs will be stored in the PostgreSQL database, how it will interact with other senior projects working with the lab, and how lab partners will be able to interact with the data compared to their older system.

1. Introduction.

This senior project will focus on organizing cookbooks containing recipes in a PostgreSQL database, with lab partners being able to query the database through a webapp managed by the Flask micro-framework. Cookbooks are made up of “meal equivalents”, with a single meal equivalent being the ingredients, amounts of ingredients, directions, and amounts of macronutrients present in the meal equivalent. When meal equivalents are compiled into a PDF document in a cookbook, relevant information is posted on recipe cards, and these recipes form the cookbook that is given to a patient or made available to the general public. Cookbooks can be prescribed to a specific patient and be tailor-made for them, or they can be assigned a “general” prescription, where the cookbook is not assigned to a specific patient, but the meal it contains revolve around a theme, whether it is a kind of cuisine such as Hispanic or Italian, dietary restrictions and preferences such as gluten-free or kosher, or the specific ingredients the meal equivalents contain.

An important feature of these cookbooks is that the data they contain must not be altered once a cookbook is sent to a patient or released to the public. This is to ensure

accurate record keeping so the lab can retrieve previous cookbooks and meal equivalents to refer to. This includes being careful with the use of foreign keys, as even nutrition facts of the same exact products can change after a few months or even a few weeks, and since this data is not consistently updated on an online database, it is up to the project to maintain records of the cookbooks, the meal equivalents, and the nutrition facts stored in the Foodomics database. This senior project will not focus on the meal equivalents nor the Foodomics data, but will rely heavily on the relevant senior projects, as it relies on accurate data from both to store and keep for records. However, even though records should not be altered after the relevant cookbook has been sent, the project should be able to grab accurate data automatically from relevant senior projects during the creation and revision of cookbooks. Relevant data can include diet prescriptions for a certain patient, the amount of fat in a certain ingredient, or whether a meal equivalent contains a certain ingredient or kind of ingredient. The collection of these attributes, some of which can be related, is relevant to an additional feature that was requested.

Since the lab has been operating since 1995¹, there are many records of meal equivalents, ingredients, cookbooks, and more, with very few ways to sort them. This problem can be seen especially when attempting to create a meal equivalent. During the summer, I created two meal equivalents to help me get an idea of how the current system works and what improvements my senior project could make. One of the most difficult parts in creating these meal equivalents was finding ingredients. The large Foodomics spreadsheet could not be sorted or filtered to show only relevant ingredients that I was looking for, and even using Excel's "Find" feature proved to be tedious. Therefore, a filtering system is sorely needed and a part of this senior project. While the cookbook project will not deal with ingredients directly, it will have to work with finding meal equivalents that fit certain parameters, most of which will be in the form of "tags". For example, a meal equivalent that is vegan will have a "vegan" tag. In the cookbooks senior project, when adding meal equivalents to a cookbook, one can search for vegan meal equivalents by using the "vegan" tag, rather than having to look at meal equivalents in depth to confirm whether they are vegan. The inclusion of these filters, as well as the other features mentioned above, bring a massive quality-of-life improvement to the system, and

will help lab partners quickly create cookbooks so they can focus more time conducting research and bettering the system.

1.1. Problem Domain.

While the construction of the diet prescriptions and meal equivalents themselves are not directly related to computer science, the systems that store and calculate this information is. Currently, data is stored in massive Excel spreadsheet which take a long time to load and do not share data with each other, meaning if data is changed in the Foodomics spreadsheet, for example, a lab partner needs to manually update the relevant information in other spreadsheets. In addition, as mentioned before, it is difficult for lab partners to find information, as it is difficult to sift through data and filter data. With these problems in mind, using a database rather than Excel spreadsheets is the first step to solving these problems. When creating a meal equivalent, a lab partner refers to a copy of the Foodomics spreadsheet attached to the meal equivalent spreadsheet, a copy which could be out of date. In addition, since the spreadsheet uses Excel equations to compute values and the spreadsheet needs to be edited directly, a lab partner may accidentally delete an important equation when editing the spreadsheet. By using a database, there is no longer a need for local copies of spreadsheets and having to copy information over, as information can be referenced with the use of foreign keys. In addition, the equations used to calculate certain values are kept separate from the database and cannot be edited by a lab partner. By using a PostgreSQL database and the SQL querying language, the data can also be filtered more easily, with queries being able to filter cookbooks by the patient it is prescribed to, the diet prescription, and more. In addition, a web-based frontend is needed for lab partners to interact with the database, which is where Flask and Flask-SqlAlchemy allow a webapp to communicate with a database, allowing lab partners to view and edit database entries through the frontend rather than having to learn SQL and attempt to work with data that way.

1.2. Previous Work (Literature Search).

The lab originally started in 1995, with the first article about the lab that I could find being published in 2006¹. This paper demonstrates the development of an evidence-based ketogenic therapy program, the same program in use today. Other papers published such as *Refinement of tools used to determine outcome of nutrient therapy for seizures*²

showcase refinements and updates to the ketogenic diet prescription system, which help illustrate for me and other senior projects students the main ideas and important details about the lab. In short, these papers demonstrate how the diet prescriptions are calculated, how the ingredients are selected, how the meal equivalents are calculated based on the previously mentioned information, and how the cookbooks are compiled and sent to patients.

This project had me dealing with new frameworks and systems that I was unfamiliar with, namely PostgreSQL and Flask. This led me to look at not only online documentation of both technologies³⁻⁵, but also led me to look at published materials⁶ where I could learn how to most effectively utilize these technologies to my advantage. While the published materials are somewhat out of date, they did provide valuable information in examples and explanations of both technologies, such as how to utilize Flask and flask-SqlAlchemy to connect to a PostgreSQL database and execute SQL queries with Python code. In addition, a previous lab partner provided a small tutorial in the form of a GitHub repository⁷ which also provided me with examples to refer to when building my own project.

2. Technical Approach (Solution).

To allow lab partners to create, view, and edit cookbooks, I began with collaborating with the other seniors working on senior projects as well as Dr. Borum and the lab team to create an ER diagram that can accurately describe the solution we wanted to implement. The Figure below shows the representation of the table and the frontend for the cookbooks senior project, but the full ER diagram shows this in context with the other projects, as well as what each of the other projects will provide and require to and from the other senior projects.

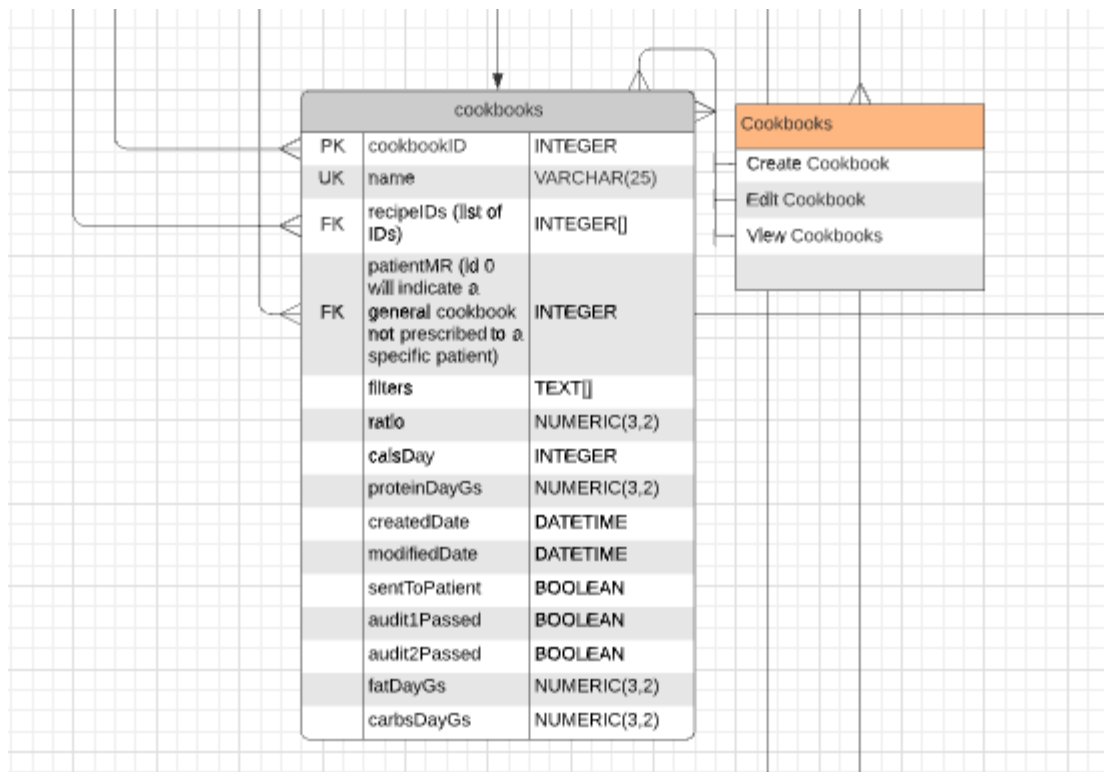


Figure 1. Section of the ER diagram connecting all senior projects involved with the lab, focused on the cookbooks table and representation of the cookbooks frontend.⁸

In order to make sure that lab partners would be able to work with the database with less difficulty and more integrity than working with the excel sheets directly, I began working on the frontend that would allow lab partners to interact with the database. I began by creating wireframes in Figma, frequently presenting my designs such as the one in the Figure below to the lab partners, changing the designs to not only fit with the other senior projects but also to accommodate the lab partners, such as by displaying relevant information, hiding irrelevant information, and providing an easy-to-follow display that would allow lab partners to quickly figure out how the system would work. In addition, I made sure that the webapp would be accessible to all users by using easy-to-read colors and background colors, using labels for inputs, and grouping relevant options together, providing an easy-to-understand user experience that was also accommodating.

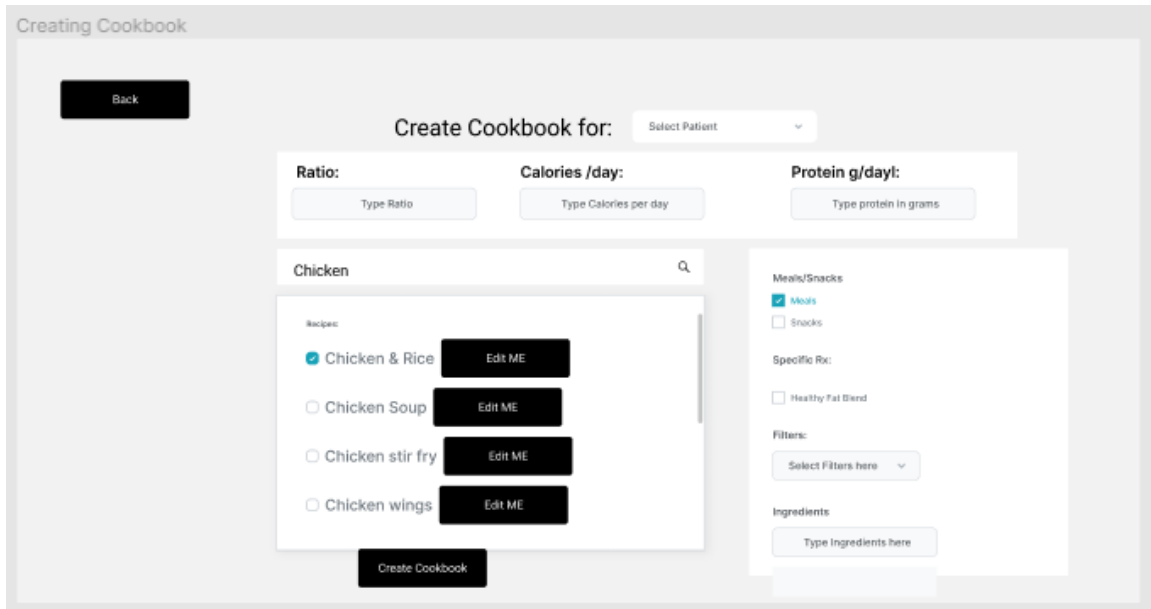


Figure 2. Figma Wireframe representing the “Creating Cookbook” page

After gaining approval for both the proposed designs for the front and back end, I began to work on the Flask app, using the previously mentioned GitHub repository⁷ as a guide to how the webapp should be structured in terms of its files. This is where the first major hiccups in the project began, as there were very few resources available for working with Flask and SQLAlchemy. Thankfully, the resources I did find provided helpful examples and tips to help me work on the project and add features. I first started building a webapp with “dummy code”, using fake recipe names such as “meal equivalent 1” to get an idea of how the project would look like without worrying about connecting the project to PostgreSQL yet. Then, when I was satisfied with how the frontend looked, I began to work with connecting a sample PostgreSQL database to my Flask app and use data present in it to simulate how real data would be obtained, presented, created, and edited. This portion of the project was also difficult due to the lack of resources, but I was able to successfully implement features that worked with the PostgreSQL database.

The frontend is structured as such: the index page (home page) will provide the user with two options: either to view existing cookbooks or to create an entirely new cookbook. When the user clicks on the “Create” button, they are taken to a different page with a form like the wireframe in Figure 1. There, the user can select the patient the cookbook is for, input the patient’s diet prescription (the ratio of grams of fat to the sum of grams of protein

and grams of crabs per 100 grams, Calories per 100 grams, grams of fat per 100 grams, and grams of protein per 100 grams). Below these inputs is a list of meal equivalents, as well as additional filters to filter the meal equivalents by, such as if the meal equivalent is a “meal” or a “snack”, if the meal equivalent is a “Healthy Fat Blend”, and text boxes to filter by ingredients and other filters. When the user inputs these values, the webpage will update the list of meal equivalents to only display meal equivalents that fit the values currently entered. To add meal equivalents to the cookbook, the user clicks on the corresponding checkbox next to the meal equivalent. When the user is finished, they can save their work by clicking the “Submit” button, which will save their entries into the PostgreSQL database

If the user clicks on the “View/Edit Cookbooks” button, they are greeted with table of cookbooks which can be sorted by date, by patient, and by filters which can be entered in a text box. Like on the “Create” page, this table will update as data is entered. Each row represents a cookbook, and the columns contain the cookbook’s name, the date it was created, if it has been sent to a patient, and the diet prescription values mentioned before. In addition, each row contains three buttons: view and edit, which allows the user to view or edit the cookbook by clicking on one or the other respectively, and download, which allows the user to download a pdf of the cookbook in the format currently used by the lab. If the cookbook is marked as “sent” to a patient, the edit button is disabled, as you cannot edit a cookbook that has already been sent to a patient. The PDFs of created cookbooks are created with existing templates of recipe cards currently being used by the lab which are edited with the use of pdf-lib⁹, a JavaScript library that allows the use of code to create and edit PDFs.

If the user chooses to view the cookbook, they will see a page displaying all the relevant information about the cookbook, including a list of the meal equivalents in the cookbook, the patient the cookbook is prescribed to, and the diet prescription the cookbook is following. There are also a few other options that were not available when creating the cookbook, namely the “Audit 1”, “Audit 2”, and “Sent to Patient” checkboxes. Before a cookbook is sent to a patient, the cookbook is audited twice by the lab team, with these checkboxes marking the audits the cookbook has passed. In addition, the cookbook should indicate whether it has been sent to a patient so lab partners know they cannot edit the

cookbook, which is what the “Sent to Patient” checkbox is for. On the top right of the screen, there is an “Edit” button, which the user can click to edit the cookbook. This leads to the same page as the “Edit” button on the respective row on the previous page. As before, if the cookbook has been marked as “sent” to a patient, the edit button will be disabled.

If the user chooses to edit the cookbook, they will see a page like the “Create” page, with a few key differences. By default, the meal equivalents already added will be displayed on a separate list to the left, with a list on the right displaying unselected meal equivalents. In addition, options to check or uncheck the “Audit 1”, “Audit 2”, and “Sent to Patient” buttons are available. When the “Sent to Patient” option is checked, an alert will pop up letting the user know that if they were to submit their changes, they would be unable to edit the cookbook. As before, if the user wants to save their work, they can click the “Submit” button, which will return the user to the index.

This project will be part of a larger web application that houses all the senior projects for this semester, including the meal equivalent and Foodomics projects. The only direct link to a different project will be when creating or editing a cookbook, the user will have the option to view or edit a meal equivalent by clicking the corresponding button, which will lead users to the meal equivalent project. This web application will only be accessible to lab partners, and they will be able to send the downloaded cookbooks to patients via email, as they have done previously. The Figure below shows an example of part of a recipe card, which is the template for downloaded cookbooks.

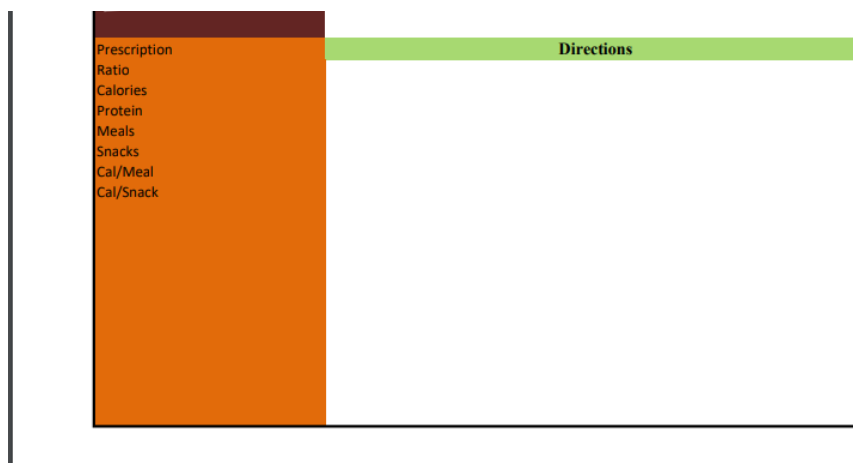


Figure 3. A section of an empty recipe card created with pdf-lib

3. Results

The project is still undergoing its final stages of development and debugging, so results on the final project have yet to be obtained. However, feedback on unfinished stages of the project has been mostly positive, with lab partners who have tested and demoed the wireframes and early versions of the Flask app reporting that for the most part, it was easy to understand, although there were some features that were either confusing or were not needed. After lengthy discussion with these lab partners as well as with my advisor, we were able to clearly communicate what was supposed to be developed rather than what was developed, which prompted me to implement changes necessary to alleviating the lab partners' issues.

4. Acknowledgements

The author would like to thank his advisor, Dr. Peggy Borum, for her guidance, advice, and encouragement toward successful completion of this project. Despite our miscommunications and frustrations, it was a joy to work on this project, especially with working on a project that helps people living their best lives. I would also like to thank Octavio Ochoa, for without his help in getting me acquainted with Flask and PostgreSQL, this project would not have been possible.

5. References

1. L. E. Little, M. H. Layman, and P. R. Borum, "Developing an evidence-based ketogenic therapy program," *FASEB J.* **20**(4), A183–A184 (2006) [doi:10.1096/fasebj.20.4.A183-d].
2. H. Allen and P. Borum, "Refinement of tools used to determine outcome of nutrient therapy for seizures (LB367)," *FASEB J.* **28**(S1) (2014) [doi:10.1096/fasebj.28.1_supplement.lb367].
3. "Welcome to Flask — Flask Documentation (2.0.x)," <<https://flask.palletsprojects.com/en/2.0.x/>> (accessed 11 October 2021).
4. "Flask-SQLAlchemy — Flask-SQLAlchemy Documentation (2.x)," <<https://flask-sqlalchemy.palletsprojects.com/en/2.x/>> (accessed 11 October 2021).
5. "PostgreSQL 14.1 Documentation," PostgreSQL Documentation, 11 November 2021, <<https://www.postgresql.org/docs/14/index.html>> (accessed 29 November 2021).
6. H.-J. Schonig, *Mastering postgresQL 10 expert techniques on postgresQL 10 development and administration*, 1st ed., PACKT Publishing, Birmingham (2018).
7. O. Ochoa, *Flask Tutorial*, HTML (2021).
8. "Borum Lab ER Diagram (UML notation): Lucidchart," <https://lucid.app/lucidchart/6aec8114-da24-43d4-ad57-5e580879d008/edit?viewport_loc=-1875%2C2253%2C2067%2C930%2C0_0&invitationId=inv_85ee82bc-d231-4079-806f-3ce72f6ceb38> (accessed 29 November 2021).

9. “PDF-LIB · Create and modify PDF documents in any JavaScript environment.,” <<https://pdf-lib.js.org/>> (accessed 29 November 2021).

6. Biography

Adam Weizman was born in Aventura, Florida on July 23, 1999, and grew up in Hollywood, Florida. He completed his secondary education at South Broward High School, receiving an AICE International Degree and completing numerous AP classes. He is currently completing his baccalaureate degrees in both Music and Computer Science at the University of Florida (Gainesville, FL), where he expects to graduate on December 18, 2021. Mr. Weizman currently has a job offer at the company Reynolds & Reynolds as a software engineer, where he plans to begin working at their Tallahassee office in January 2022. He has completed multiple internships this past summer, working on the websites for Ancient Flow Therapy and Katie White Consulting & Design, LLC, two local Gainesville companies. In addition, he was a design coach for DC Design’s summer program DC Design the Future, a program that tasks high school students with designing and building projects for users with disabilities. Mr. Weizman was employed as a design coach, where he coached a team of high school students in designing a mobile app for a user with migraines to help log their symptoms. Mr. Weizman is an avid programmer and musician, having years of experience in both computer programming and piano playing. He enjoys music, video games, game design, languages, and football, and loves when music intersects with computer science, which is why he hopes to work as an audio programmer in the future.