

# Foodomics Database Senior Project

Devin J. Blem

CIS4914, Senior Project  
Department of CISE  
University of Florida

Advisor: Dr. Peggy Borum, *email:* prb@ufl.edu  
Department of Food Science & Human Nutrition  
University of Florida, Gainesville, FL 32611

Date of Talk: 6 Dec 2021

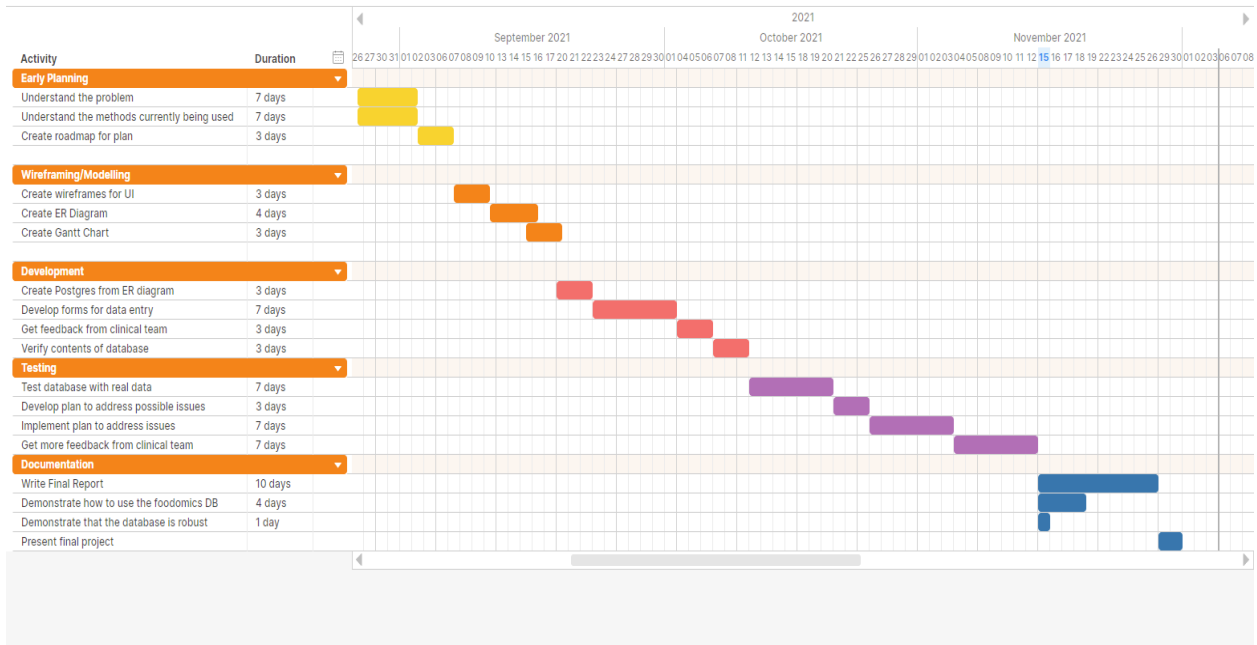
## **Abstract**

The goal of this project is to create an environment and SQL database that members of Dr. Borum's lab can use to interact with what is known as the Foodomics Database. The Foodomics Database provides detailed nutritional information for thousands of different food products. Currently, the database is contained in an excel spreadsheet that has been growing for more than 25 years. As a result, the spreadsheet has become excessively large and can take a long time open, close, and sync. On top of that, the spreadsheet utilizes several convoluted excel formulas which are susceptible to accidental changes that can cause certain functions to become unusable. Lab members have also had issues trying to link the various spreadsheets together, which has resulted in the need to copy the same information into multiple spreadsheets. The system at large has become increasingly complex over the years, and they have decided that their future plans would be better served with a relational database rather than the current spreadsheet system.

In order to address these problems, we have created an application based on the python framework Flask. The application will connect to a PostgreSQL database and allow lab members to better analyze and manage nutritional data as well as patient data. This project focuses solely on nutritional data but it provides the foundation for various other functions in the application, namely the diet prescription and meal equivalent calculators. The application also deals with restricted data, which means that it must comply with HIPAA laws as well as university policies regarding research data.

## **1. Introduction**

This report summarizes the development of a software system that will allow Dr. Borum's lab team to easily, effectively, and securely manage the Foodomics database. The database has been tested for robustness and the GUI has been designed in a way that lab members can quickly learn how to work with it. The software we have developed this semester has laid the groundwork for major improvements to the way that the Borum Lab will be able to administer their nutritional intervention for patients, known as Precision Ketogenic Therapy (PKT). The application's design could be applied to other database projects that deal with restricted data. It could also allow the lab to expand the program to other schools or healthcare providers in the future. Figure 1 is a Gantt chart detailing the tentative schedule for the project.



**Figure 1.** Tentative Project Schedule

## 1.1. Problem Domain

As mentioned above, the current system used by the lab involves Excel data hosted on SharePoint. This has made it difficult for lab members to perform advanced data analytics, and it is highly prone to human error. The lab has had issues where members will unknowingly make changes to a formula and causing functions to break. There have also been many issues with synchronization since the same information can be found in several places. The main topic of interest for this project is data science and software engineering. Due to the amount of data that the lab has accumulated, the solution to use a relational database with PostgreSQL was determined to be necessary for the future expansion of the PKT program. The relational database will provide much needed structure to the food and patient data, which will allow deeper and more meaningful insights to be drawn.

## 1.2. Literature Overview

Understanding the software side of the program requires working knowledge of SQL databases, HTML/CSS, Docker, and the python web application framework Flask. Docker is essential because it allows development to occur outside of ResVault. The Docker reference

documentation is a useful resource for learning how to perform tasks such as building Docker images. The project also requires a solid grasp on the python programming language in order to work with Flask. The Flask guide on Tutorialspoint contains a plethora of useful information regarding how to build a Flask application and how to utilize various functions properly. Creating the overall look and style of the application requires knowledge of HTML and CSS. The various “How To” guides on W3Schools are useful to reference for basic styling and layout tools. Familiarity with SQLAlchemy is also required since it allows the Flask application to communicate with the database. The SQLAlchemy documentation contains helpful information regarding how Flask applications talk to SQL databases. Finally, it is necessary to be familiar with PostgreSQL since the application utilizes a PostgreSQL database. The PostgreSQL documentation has an in depth review of the various data types, functions, operators, etc. that are necessary to create and query the database.

In order to understand the clinical aspect of the application, it is important to review the various internal documentation regarding proper data entry into the Foodomics Database, namely the Foodomics SOPs. The Foodomics SOPs describe how to perform a “food run” or “nutrient run” in which lab members collect information from the nutrition fact labels of various name brand goods and then enter that data directly into the Foodomics database. They also photograph the front of the items, control numbers, prices, manufacturers, and package sizes. It is crucial that data entry is done properly in order to ensure the integrity of the database, and therefore it is critical that lab members review the Foodomics SOPs thoroughly before entering data. These documents are located in the lab’s Microsoft Teams/SharePoint files.

## **2. Technical Approach (Solution)**

In order to better serve the complexity of the lab’s research in the future, it was first determined that a relational database would be better suited for the task than the current flat spreadsheet system. In the last year, lab members established a relational database to accommodate the output data of their patients. However, a relational database had not yet been established for the input side of the data, which includes the Foodomics Database. Therefore, the first step of the modelling process was to create an ER diagram for the contents of the nutrition facts database. Selecting the proper data types to represent the various columns of the Foodomics Database and

determining the relationships it has with the other aspects of the application was a critical part of the design process. The ER diagram for the foodomics database is a single entity that has a one-to-many relationship with the diet prescription and meal equivalent calculator, which are two major tools that the lab uses to administer PKT. The actual ER diagram will be omitted from the report due to its size, but it provides relatively complete molecular compound information for the various food products. The columns are currently 1:1 with the current spreadsheet database, which is required for exporting the spreadsheet data into PostgreSQL.

Next in the modelling process was to create wireframes for the user interface. These wireframes represent what lab members would see once they have navigated to the Foodomics section of the application. Figures 2 and 3 are wireframes for the new data entry page and the page for viewing the database itself, respectively.

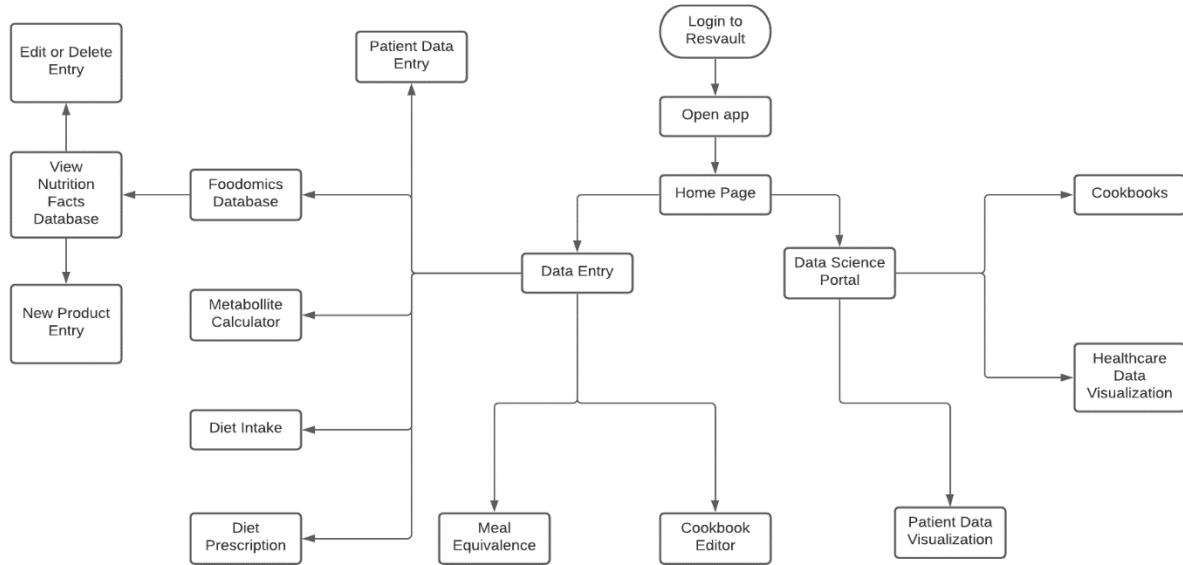
The wireframe shows a web browser window titled "A Web Page" with a URL bar containing "https://". The main content area is titled "New Product Form". It contains five vertically stacked input fields with the following labels: "NDID", "Product Name", "Source", "USDA Food Group", and "Sub Food Group". At the bottom of the form are two buttons: "Submit" and "Cancel".

**Figure 2:** New product entry form

The wireframe shows a web browser window titled "A Web Page" with a URL bar containing "https://". The main content area has a search bar with a magnifying glass icon and the text "search", and a filter icon. Below these is a table with four columns: "NDID", "Product Name", "USDA Food Group", and "Sub Food Group". Each column contains four rows of placeholder text "xxxxxxx". Below the table are two buttons: "Add a Product" and "Edit on existing Product".

**Figure 3:** Viewing the database

The final step in the modelling process was to create a flowchart to demonstrate the steps that a lab member would take to get to the Foodomics Database. Figure 4 is the flow chart for navigating the application, which begins with logging in to ResVault.

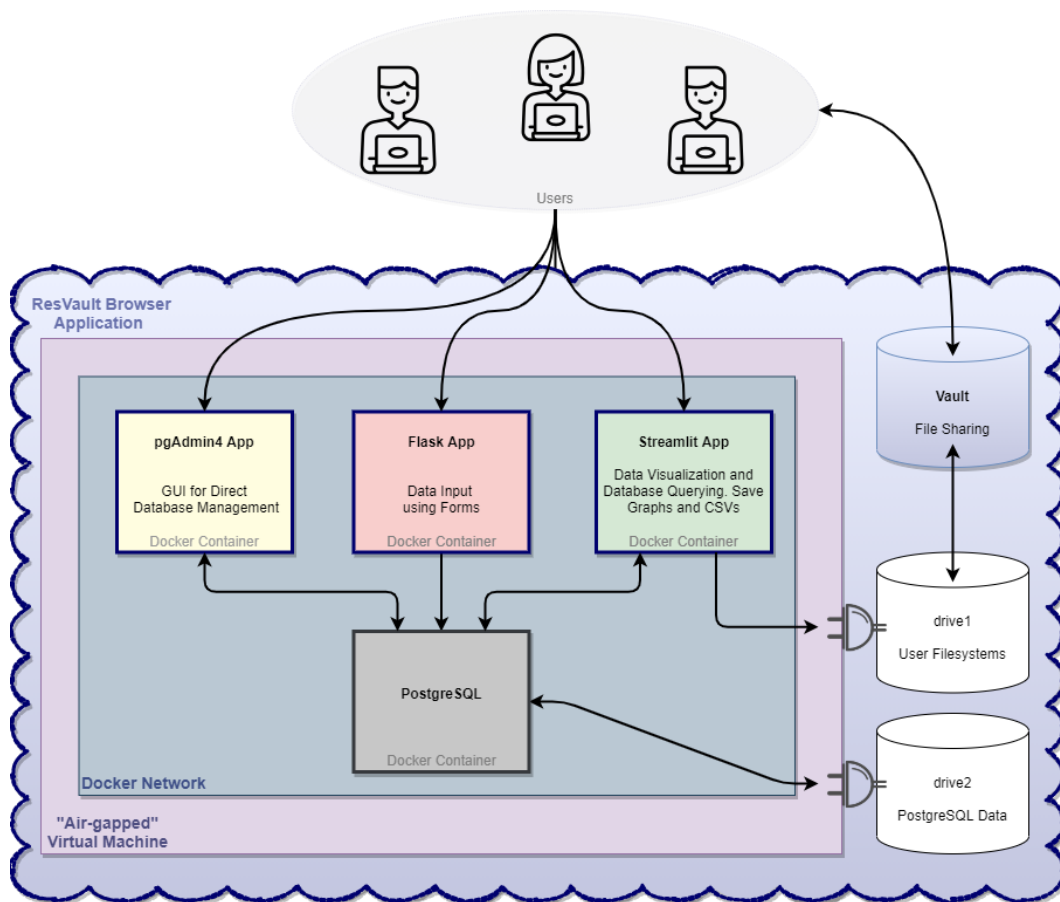


**Figure 4:** Application flow chart

Once the modelling phase of development was completed, it was time to begin programming. The first part of the programming process was establishing the new database. This involved several steps, beginning with creating the SQL table for the nutrition facts database. Once the SQL table was complete, the next component was to create a “map” from python to the new PostgreSQL database using SQLAlchemy so that the application and database can communicate. After the database was established, it was time to develop the interfaces for entering new products and viewing/editing the contents of the database. Several columns in the foodomics database are calculated based on other columns, and these calculations utilize excel-style formulas. These calculated columns required equivalent formulas to be developed in python.

A key aspect of our design approach is centered on data security. Since the PKT software deals with restricted research data, it must comply with HIPAA laws and university policy regarding research data. As such, we will deploy the application using UF’s ResVault. ReseachVault (ResVault) is a secure computing environment that allows for research to be conducted on restricted data, specifically projects requiring compliance with NIST 800-171 and

NIST 800-53 standards. ResVault works as a browser application that is used to access files in a "Vault" and virtual machines that are "air-gapped" or disconnected from outside networks. The software in this repository is hosted with a virtual machine, which includes a Postgres Database, and two applications that run on python frameworks, Flask and Streamlit. ResVault also provides drives that plug in to the VM. Figure 5 is a diagram of what this system looks like.



**Figure 5:** ResVault browser application diagram

Because the VM does not have internet access, it comes stripped of software and developers cannot install necessary packages on it directly. To allow development outside of ResVault, we use Docker. Any development that is done outside of ResVault can be packaged into Docker images, uploaded to ResVault, and used to spin up Docker containers that host the applications in a Docker network.

### **3. Results**

The software has shown to be robust when tested with real world data. When testing the new application with known input values from previous nutrient runs, the application has produced calculated output values that match those of the old system. Early results have been promising, but the system should still undergo more thorough testing to ensure that there are no edge cases which could potentially create conflicts or errors in the program and its functions.

### **4. Technical Challenges and Solutions**

One expected difficulty going into the project process was a lack of experience with full stack development. When working with an unfamiliar framework, it is important to utilize the numerous resources that are available online. The references below, which were discussed in the literature overview, provided the information required to build a firm understanding of how the application works and how it was created.

Another challenge that new members of the lab can face is unfamiliarity with food science and the processes involved with data collection. As mentioned in the literature overview, it is critical for lab members to perform the data entry process properly. The best way to do this is to perform a “food run” or “nutrient run” of your own after reviewing the Foodomics SOPs.

### **5. Concluding Remarks**

In closing, this report has established the technical approach and system design for a Flask application that can interact with the new relational Foodomics Database in PostgreSQL. From drawing wireframes and flowcharts to creating databases and the forms that interact with them, this project has been a crash course on full stack development and data science. An experience like this is invaluable for aspiring software engineers.

This project is the beginning of a long term goal of the Borum Lab to have a free and open foodomics database for anyone to access. The database is constantly updated as items are always changing. One major area of improvement would be to implement a better system for determining which food products need a nutrient run or audit the most. There are also still columns that could potentially be removed from the database which are no longer required for data analysis.



## 6. Acknowledgements

The author would like to thank Dr. Borum for her involvement in overseeing the project from start to finish and for all of the wonderful work that the Borum Lab does for its patients.

## 7. References

[1] PostgreSQL 14.1 Documentation, The PostgreSQL Global Development Group. [Online]

Available: <https://www.postgresql.org/docs/14/index.html>

[2] Docker Reference Documentation, Docker Inc. [Online]

Available: <https://docs.docker.com/reference/>

[3] Flask Tutorial, Tutorialspoint, 2021. [Online]

Available: <https://www.tutorialspoint.com/flask/index.htm>

[4] W3Schools How To, W3Schools. [Online]

Available: <https://www.w3schools.com/howto/default.asp>

[5] SQLAlchemy 1.4 Documentation, SQLAlchemy, 2021. [Online]

Available: <https://docs.sqlalchemy.org/en/14/>

## 8. Biography

Devin J Blem was born in Plantation, Florida on April 4, 1997. He completed his secondary education at Pompano Beach High School and is currently finishing a degree in Computer Science at the University of Florida. Mr. Blem hopes to enter the software engineering when he graduates and is proficient in C++, Java, and Python. He also has experience with full stack development and databases thanks to his curriculum at UF and time spent developing software for the Borum Lab.