UTRITION IV

SOFTWARE DESIGN DOCUMENT

Kiera Belford, Thomas Phelan, Jared Schneider, Stuti Shah, Brittany Shine
CSE 5915
Team HealthBucks

24 April 2021

Capstone Sponsor
Andee Peabody
Founder & CEO, Engage Health

Capstone Supervisor
Robert Vanderwall
The Ohio State University College of Engineering
Dept. Computer Science and Engineering

Table of Contents

1. Outline

The long-term goal of Utrition is to enable patients and end-users with nutritional and dietary recommendations based on their genetics and overall diet.

2. Stakeholders

For our project, our primary stakeholder would be Andee Peabody and EngageHealth as they are the product owner and business sponsor for this assignment. Other stakeholders involved are the nutritionists, clinicians, and geneticists, as it is important that the Utrition Application produces correct information for the clients. Finally, the patients who use the application will also be considered stakeholders due to the fact their health is at stake, and they will be receiving personalized dietary recommendations from the project.

3. Purpose

The purpose of this project is to improve and expand upon the work of previous teams in creating an application that can be used by nutritionists, clinicians, and geneticists to assess a client's dietary needs. The end goal of this project is to have a product that takes as input a client's current diet and DNA information (such as from Ancestry or 23andMe), and outputs a new dietary plan to help curb their risk for certain conditions to which they are predisposed. Nutrition also has a secondary purpose as a Dietary Analysis tool, but since this aspect is out of the scope of our iteration, discussion about this aspect will not be heavily discussed.

Due to the pivot in our project, our primary goal is to have an accurate and query-able database for our client. The previous group had a database schema that did not correctly interconnect the essential information that is necessary in the Utrition platform. Therefore, our goal will be to redesign the schema to allow for the correct relationships for the project to function. Our other goals include creating a crosswalk mapping between datasets that allows single queries to return results between datasets, integrating the database with the web application, adding essential functionality to the web application like file upload and download, and producing a "Eat This, Not That" report for patients to download.

4. Context

The Utrition Application project is an extension of an existing Engage Health project with the same name. Our group has begun the fourth iteration of the project. Understanding the goals and scope of our project requires background knowledge on how the web app ultimately aims to function and the previously completed components.

Utrition in its completed form takes two inputs from a patient:

- 24-hour food-recall outlining the patient's food intake over the last 24-hours
- DNA information detailing, for a plethora of gene snippets called SNPs and the alleles (variations) that the patient has for each SNP.

The gene variations are then analyzed using data sourced from external databases, to determine diseases the patient may be genetically predisposed to relative to gene mutations found in the patient's DNA. Since genes control the manifestation of several diseases, regulating a mutated gene may also help regulate the disease that the gene controls.

Based on the possible diseases, food recommendations and a micronutrient report are provided to the patient through analyzing which food/nutrients help to regulate the mutated genes.

At the onset of the fourth iteration, the previous teams completed the following aspects of the application:

- Built a UI for the web application allowing users to enter a 24-hour food profile
- Built a database and loaded the database with data from external databases in compliance with the created database schema
- Formatted nutritional information from external databases to be displayed in the completed UI for the web application

Due to the genetic component of the Utrition project and to allow Utrition to function as desired, the database's schema and data are required to adhere to the relevant biological principles, namely how a user's DNA data for particular genes are analyzed to detect genetically predisposed diseases.

However, since the existing database schema design failed to match the relevant biological principles and the data within the database did not contain the correct information, the team aimed to re-design and re-populate the database with biologically sound data and design choices rather than cleaning the existing and incorrect data.

5. Design & Technical Constraints

Due to the design choices of previous iterations, the UI components of this project required ElasticSearch and Flask, and the database was created and manipulated using PostgreSQL.

Use of Entrez Programming Utilities such as ‘esearch’ and ‘efetch’ were required to collect and format all necessary data from NCBI, the external source for gene and correlated disease data. Manually downloading the information from NCBI’s website, while faster, excludes information critical to this project.

The database is also pre-loaded with information rather than dynamically fetching information based on user input. This decision was made for two main reasons: First, at the request of the sponsor. Second, retrieving gene information directly from NCBI is a rather costly and slow process and not guaranteed to work correctly when retrieving thousands of records at once. This is due to server issues on NCBI’s side. The decision to pre-load the database, while initially time consuming, allows users to receive more quickly their results.

6. Design Structure

Figure 1 below describes how the front-end web page interacts with the Utrition database. The user begins by entering their email into the web application (in the stage labeled "Login"), which checks the database for a user with that email. If the user has never entered their information before, they are prompted by the web application to enter their personal information (in the stage labeled "Account Register"). The user's personal information is then stored in the database. Otherwise, the web application moves on to the next web page.

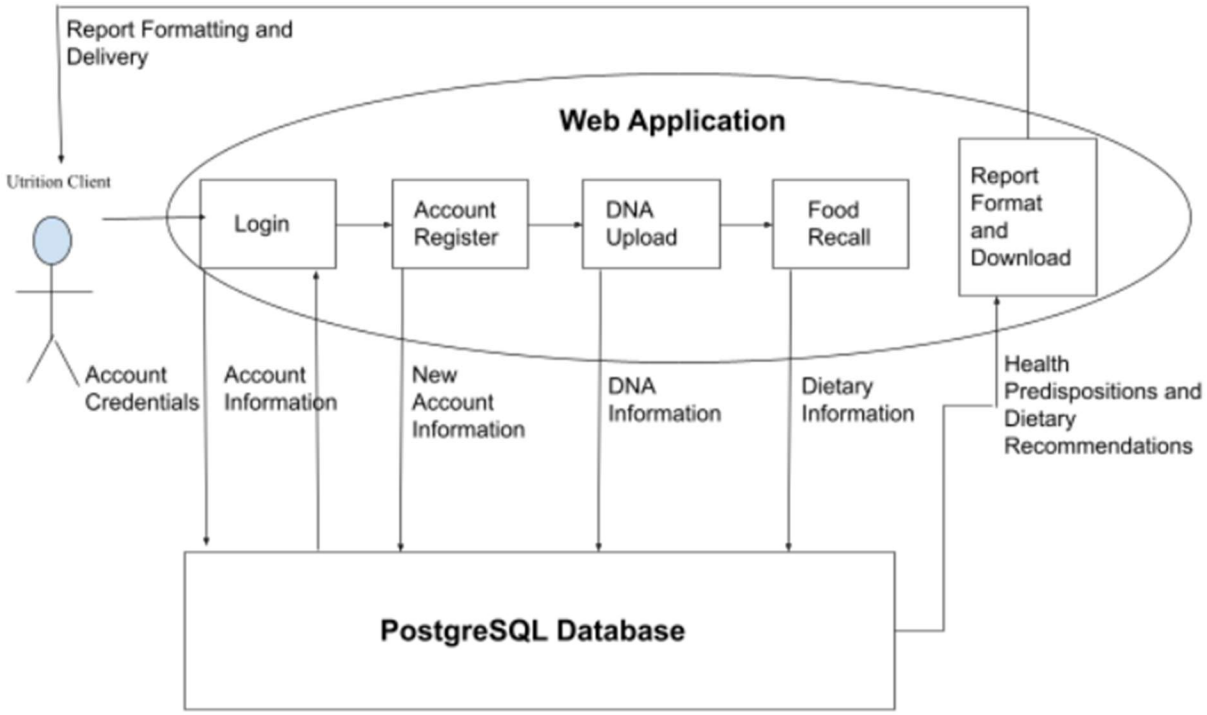

Figure 1. Utrition Web Application Block Diagram.

The next web page asks the new user to upload a file containing their DNA information (in the stage labeled "DNA Upload") which is then stored in the database or moves onto the next page if this is a returning user. Finally, the user is asked to list their daily dietary habits which is inserted, too (in the stage labeled "Food Recall").

Several queries are executed by the PostgreSQL database using the user's dietary and DNA information to create a report containing health conditions the user may be genetically susceptible to along with recommendations of which foods the user should and should not be consuming.

Note that although new users are required to upload their DNA information prior to entering their dietary habits, if a returning user's DNA information is no longer in the database, the web application will prompt the returning user to upload their DNA information prior to redirecting the user to a page to list their dietary habits.

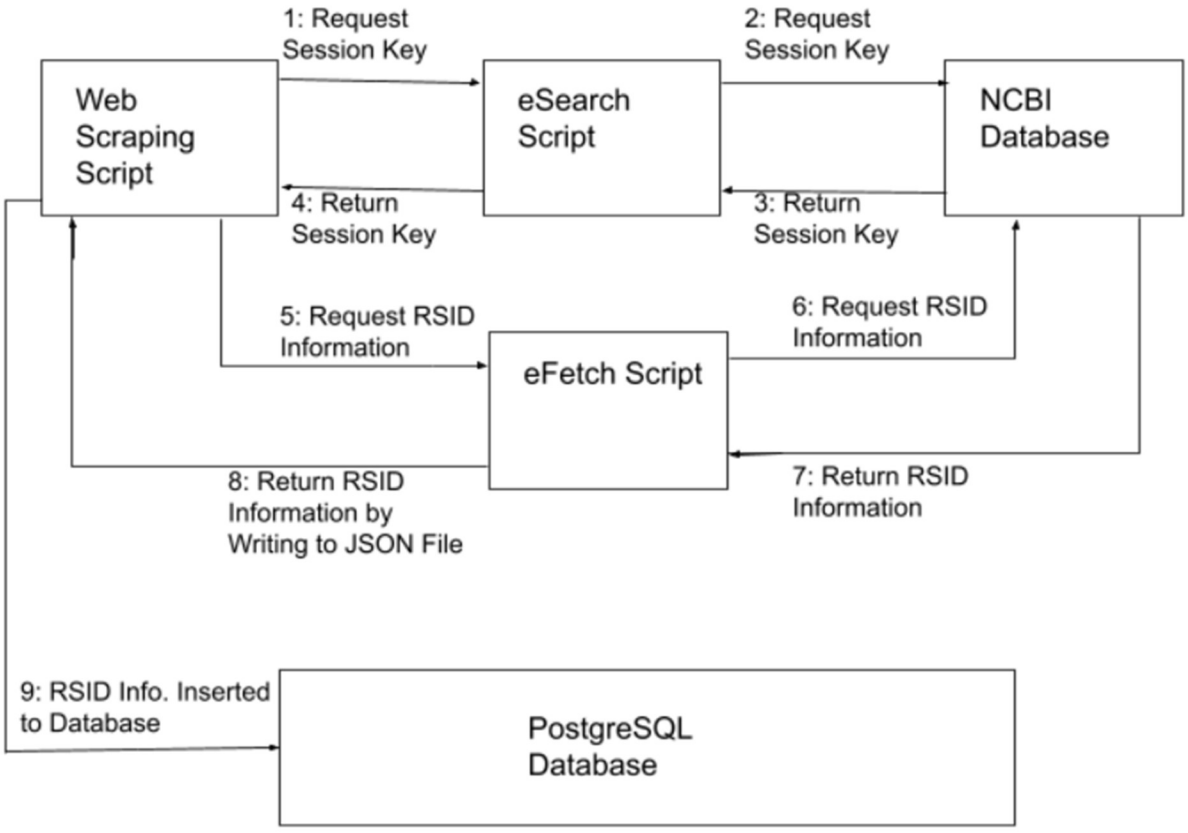

Figure 2: RSID Web Scraping Block Diagram.

Figure 2. rsID Web Scraping Block Diagram.

Figure 2 describes the process by which the data for rsIDs, mutations, and diseases are collected and stored in the Utrition database. This whole process is controlled by a bash "master script" of sorts (labeled "Web Scraping Script"). The web scraping process consists of two steps: an 'eSearch' and a repeated call to 'eFetch'. The eSearch command returns the following information:

- A web environment key
- A query key
- An ID list of records returned by the query
- An integer value referring to length of ID list

Obtaining web environment and query keys from the eSearch script ('Session Keys' in above diagram) is essential for maintaining web session information. With these keys, the team was able to repeatedly run the eFetch script in batches, tracking each eFetch start and finish index on NCBI and limiting the entire web scrape process to one eSearch command. The eSearch commands are executed in the script labeled "eSearch Script". This information is found in labels 1-4 in the above diagram.

In labels 5-8, the master script runs continuously in a loop to collect information from the NCBI database. Within the loop, the master script enters a second script ("eFetch Script") which requests and fetches information to and from the NCBI database. From each of the eFetch calls, the database returns new rsID information in JSON format and is added to a JSON-like table.

Finally, after all rsID information has been collected, the script sifts through and parses the JSON table. The parsed and cleaned data is then inserted into the correct tables and the web scrape process completes.

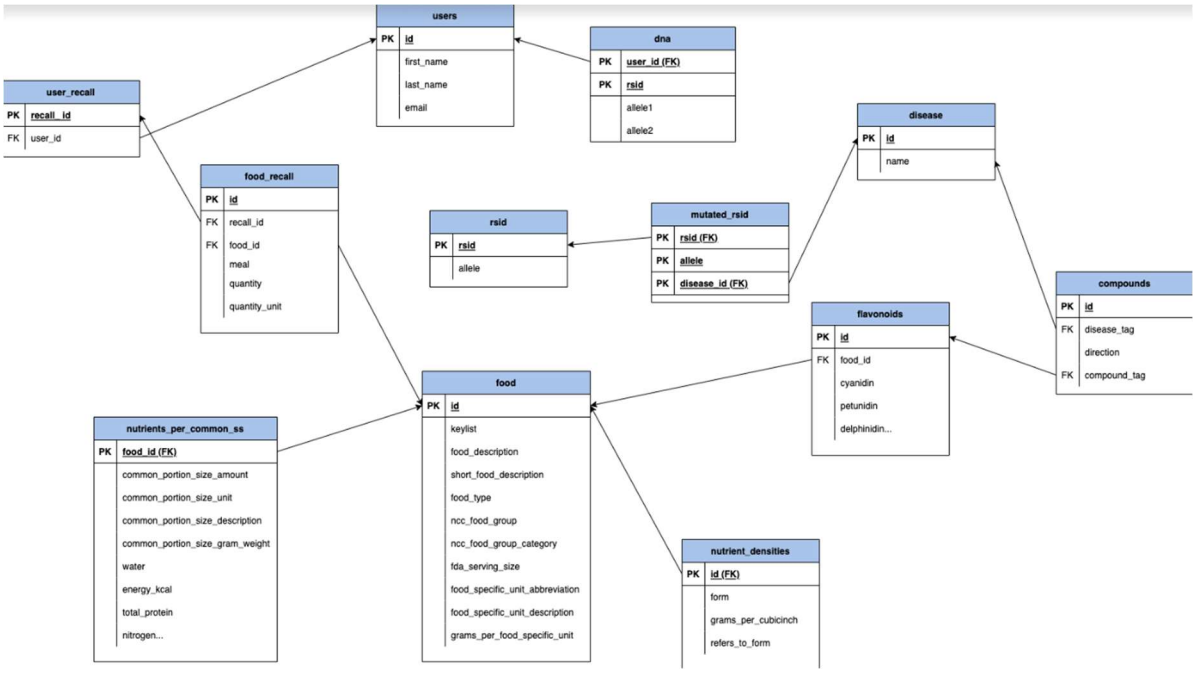

Figure 3. Revised and current database schema.

7. User Interface

The previous UI had all the necessary parts for the Utrition application. However, it lacked actual functionality. The Utrition IV team built off the original UI and fully connected the backend to the frontend for the user interface to have a working application.

Originally, the homepage would take in an email and automatically bring you to the food recall page. The team updated it so that there are now three different paths for the user cases from the homepage. These are the potential paths for a user:

1. **The user has not registered for a Utrition account.**
    - User enters a new email → they are brought to a registration page where they enter their first and last name. The user's information is stored in the database.

- Once the user is registered → they are brought to the DNA page. Here is where the user will upload their DNA information to the website.
- If DNA has properly uploaded → the user is taken to the food input page. The user can input their meals/food to the Utrition platform.
- After the user inputs their food data → it brings them to the food analysis page. The user can download the Utrition analysis. If the download fails, they are taken to a page that would look like their download. The analysis lists all foods with any flavonoids matching to a disease the user has or is susceptible to.

2. **The user has registered but has not uploaded their DNA.**
    - The user enters the email they used to register their account → they are redirected to the DNA page.
    - If DNA has properly uploaded → the user is redirected to the food input page.
    - After the user inputs their food data → it brings them to the food analysis page.
3. **The user has registered and has uploaded their DNA.**
    - The user enters the email they used to register their account → the user is taken to the food input page.
    - After the user inputs their food data → it brings them to the food analysis page.

Improvements:

- User information is now stored in the database.

- There is now a proper login/register process.
- Functionality to upload DNA files.
- Functionality to download the Utrition analysis report.
- The connection between the backend to the frontend.

8. Non-Functional Goals & Testing

Some non-functional goals for this project included:

- Straightforward and simple UI in design and use
- Informational reports that clearly outline essential information
- Readable output reports
- Database schema that was both biologically accurate yet only contained information necessary to our group or future groups.

The testing strategy for this project is to spend 50% of the testing resources on unit testing, followed by 30% for integration testing and 20% on UI testing.

Since the project heavily relies on crosswalk mapping between datasets and cleaning the aforementioned datasets, ensuring that changes to individual datasets do not impact the validity of queries is of high priority.

Unit testing for this project will include data quality tests as the team collected data from multiple sources and look to ensure that each table has the correct data in it.

The integration testing will be more heavily focused on the interactions of tables. Since a majority of our project has pertained to database building, this is the highest priority for Utrition IV to evaluate in test coverage. Utrition IV is currently unable to automate UI testing, so UI testing will be completed manually, focusing on testing both the functionality

of the UI and its components as a unit, along with testing user input and ensuring that the system only permits the user to enter valid inputs.

An obstacle faced during this project was the lack of clarity in documentation of setup from previous teams. The Utrition IV team has added to the current manual to allow for future teams to be able to start more efficiently and continue work on the project.

9. Timeline

Listed here is the Utrition IV team's timeline of deliverables and presentations for the capstone project:

- Demo 1: February 25th, 2022
- Demo 2: April 8th, 2022
- Demo 3: April 25th, 2022
- Final Deliverables: April 25th, 2022