



ACS5423: Software Development for Web – Project Phase 1

Dewayne Hafenstein

University of Oklahoma – College of Engineering

Table of Contents

Abstract	3
Design.....	4
Use Cases.....	5
Data Model	8
Wireframes (UI)	9
Structure.....	10
GitHub Repository.....	11
CI/CD Pipeline	11

Abstract

The project for ACS5423 was to develop a web application using the MEAN stack (Mongo, Express, Angular, and Node.js) technologies to explore an extract of the FDA's (Food and Drug Administration) branded food database. This database extract was loaded into a Mongo database cluster and accessed by a Node.JS server to support a browser-based UI, all developed using node and html.

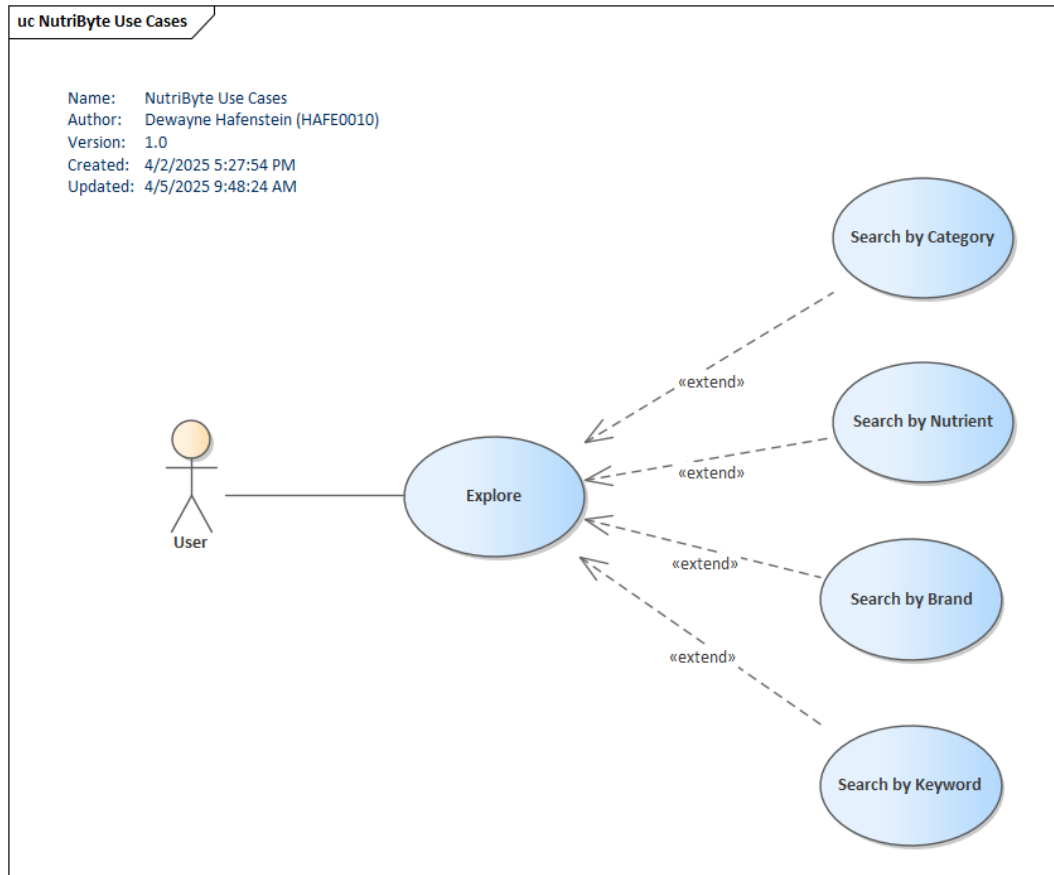
ACS5423 – Project Phase 1

This report details the design, structure, and implementation details of the project. Even though some deliverables were not requested until phase 2, they have been added as part of phase 1 and will be extended and improved in phase 2. These include the use of CI/CD pipelines and the utilization of a GitHub repository. This was done because that's the way I've gotten into the habit of software development; always start by creating a repository to manage your deliverables.

Design

Design was performed using a professional architecture design tool which I happen to have a license for, named **Enterprise Architect** by Sparx Systems, Ltd. This tool is used heavily at AT&T and I am very familiar with its use. I did not, however, build out all of the artifacts that I would normally do in a project for work. Instead, I focused on the use cases, the data model, and the UI wire frames. A detailed report from the tool is provided and is titled "**NutriByte Design Report.docx**". This report is generated from the tool and includes all the diagrams as well as comments, descriptions, and linkages between the various design artifacts. I have extracted those diagrams and inserted them into this document as well.

Use Cases




The use cases were designed to provide 4 different ways to query and obtain information from the database. These allow a user to:

1. Search the database by the categorization of the branded food. This returns a list of all foods that are categorized with the selected category. The categories are obtained from the database and used to fill a select list so the user can just pick one that already exists and need not type anything in.
2. Search by nutrient. This allows the user to search the database for all foods that contain the specified nutrient. Again, the selection list is pre-populated with the list of all nutrients that exist in the database.

3. Search by brand. This view allows the user to search the database for all foods that are produced by a specific brand, or marketed under the desired brand. Again, the list of all brands that exist in the database is used to pre-populate the selection list. This list is very large, and takes a few seconds to draw the selection list. However, the list is sorted, and the user can scroll to the brand by pressing the letter corresponding to the brands first letter to rapidly get to the appropriate grouping.
4. Finally, the ability to search the foods using a keyword search. This searches not only the ingredients, but also the nutrients, brands, and descriptions of all branded foods and returns all foods that contain the specified keyword in any of those fields.

Once the list of foods has been returned, the user can select any of the foods to obtain detailed information about that food. For example, if the user selects the view by category (tab), and selects the category “Biscuits/Cookies”, the following (partial) display results. Scroll bars are used to allow the user to scroll through the contents. If the user selects any food, a detail area expands under the food to show the detailed information about that food.

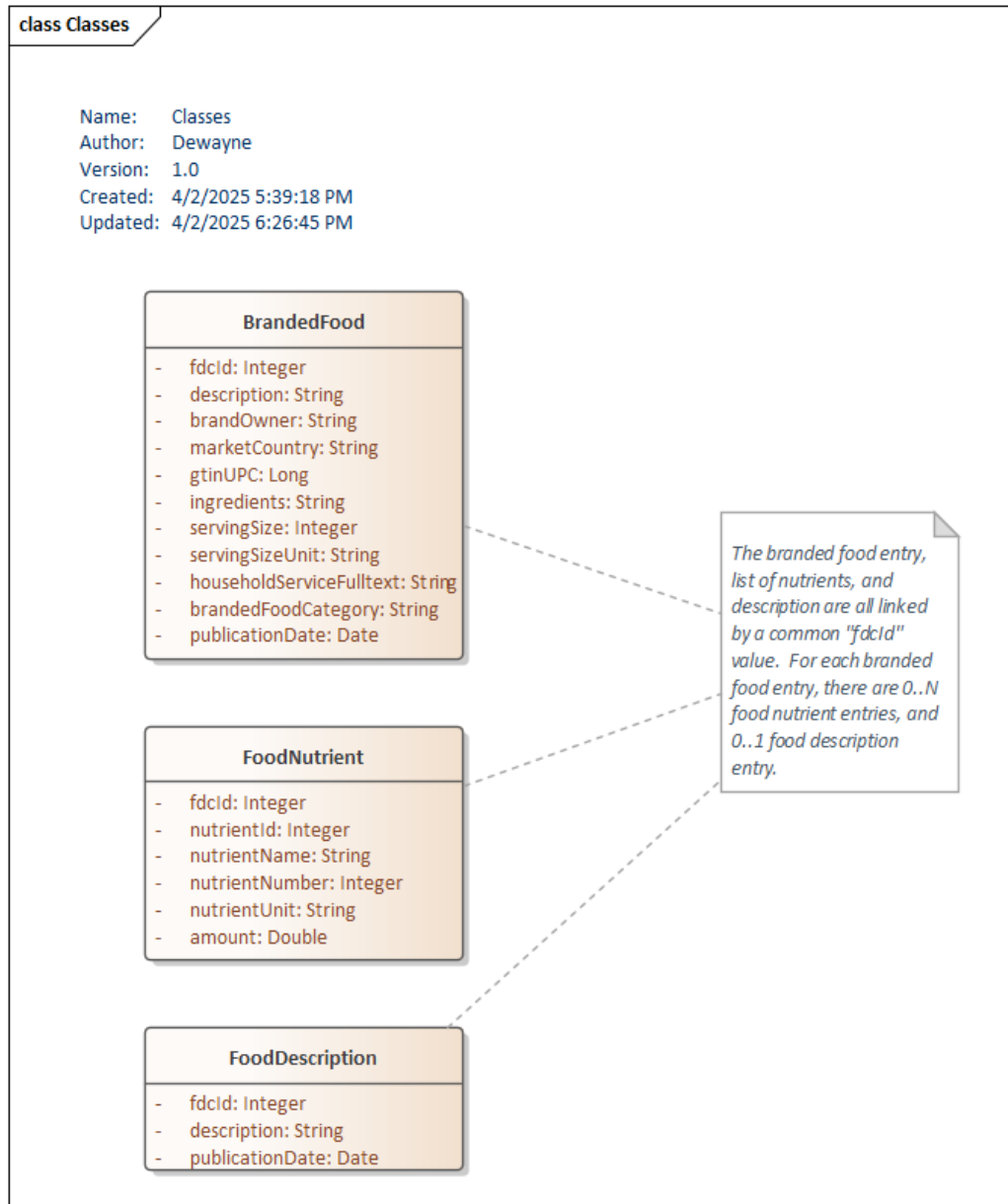
More than one food may be expanded to see the detail and to compare different foods if desired. A screen shot of the application is shown below:



Category	Nutrient	Brand	Search
Category: Biscuits/Cookies			
<ul style="list-style-type: none"> 300 LADY FINGERS FROZEN Carr's Crackers Rosemary 5oz Carr's Crackers Salt And Pepper 5.3oz Carr's Crackers Mixed Seed 5.3oz Fiber One Chocolate Chip Crunchy Cookie <div> Brand: GENERAL MILLS SALES INC. Ingredients: SEMISWEET CHOCOLATE CHIPS (SUGAR, CHOCOLATE LIQUOR, COCOA BUTTER, MILKFAT, SOY LECITHIN, ARTIFICIAL FLAVOR), WHOLE WHEAT FLOUR, MODIFIED WHEAT STARCH, BUTTER, SUGAR, BROWN SUGAR, CHICORY ROOT EXTRACT, VEGETABLE OIL (PALM, CANOLA), EGGS, SALT, BAKING SODA, MILKFAT, SOY LECITHIN, NATURAL FLAVOR, NONFAT MILK. Serving Size: 26 g Published: 2/27/2020 </div> <ul style="list-style-type: none"> Nature Valley Honey Biscuits Annie's Cheddar Squares Baked Cheese Crackers, Made with Organic Wheat Annie's Organic Honey Bunny Grahams <div> Brand: GENERAL MILLS SALES INC. Ingredients: ORGANIC WHEAT FLOUR, ORGANIC CANE SUGAR, ORGANIC EXPELLER-PRESSED SUNFLOWER OIL, ORGANIC CORN FLOUR, ORGANIC WHOLE WHEAT FLOUR (GRAHAM FLOUR), ORGANIC HONEY, CALCIUM CARBONATE, BAKING SODA, SEA SALT, MIXED TOCOPHEROLS (VITAMIN E) TO PROTECT FLAVOR, NATURAL FLAVOR. Serving Size: 28 g Published: 2/27/2020 </div> <ul style="list-style-type: none"> Annie's Organic Lemon Drop Cookie Bites Old El Paso Crispy Flour Tortilla Shells 8 Count Annie's Organic Honey Bunny Grahams Old El Paso Bold Nacho Cheese Flavored Stand 'n Stuff Taco Shells 30 Count Annie's Organic Chocolate Chip Cookie Bites 36 Count Annie's Gluten Free Baked Cracker Cheddar Bunny Tails BELVITA BREAKFAST BREAKFAST SNACK CRUNCHY HAZELNUT OREO COOKIES VANILLA 176 GR LIGA MILKBREAK BISCUITS ORIGINAL 245 GR JACOB'S MEDITERRANEO BREAD CRACKERS OLIVE OIL AND OREGANO CHIPS AHOY! COOKIES SUPERMIX 300 GR 			

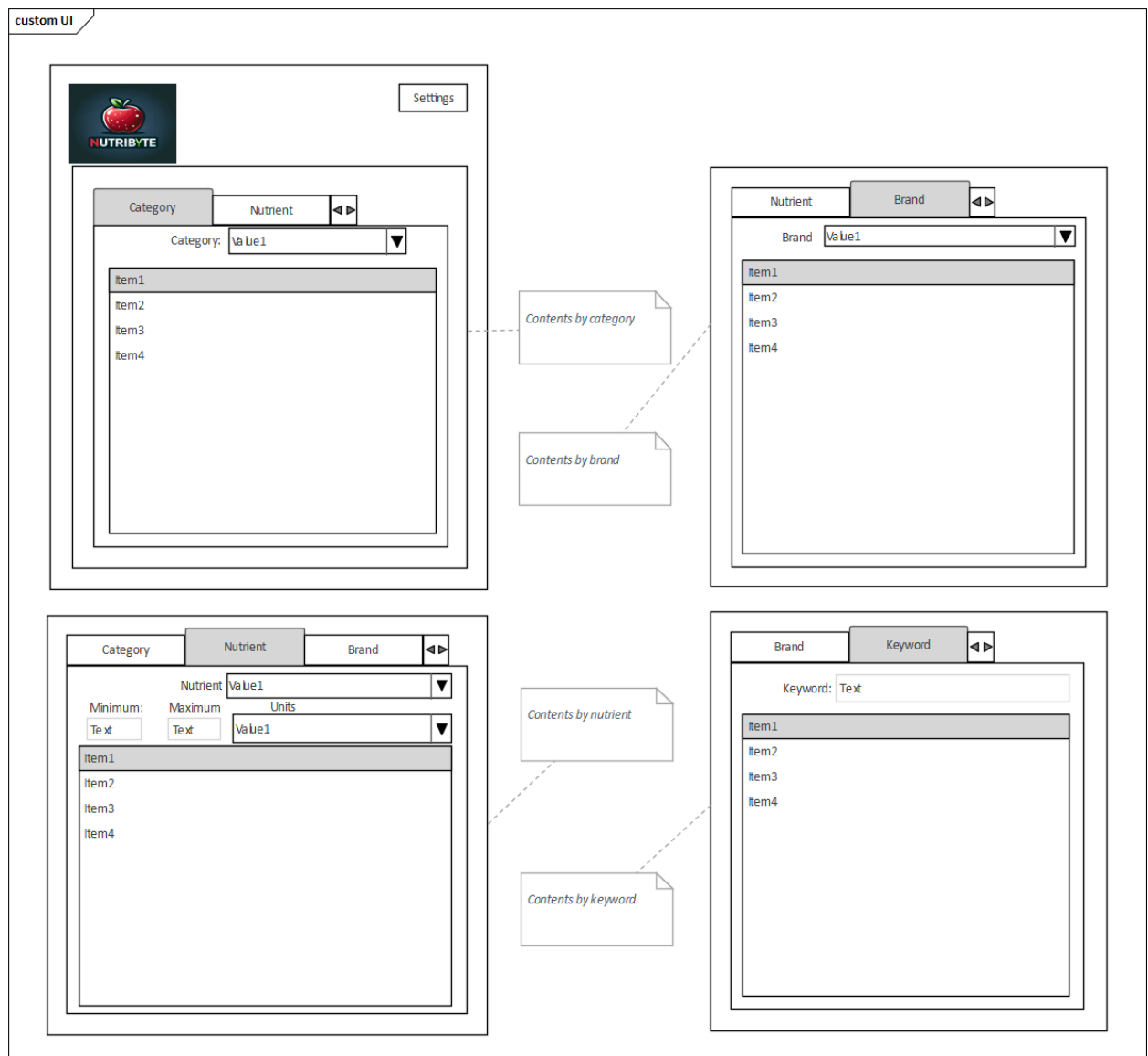
Data Model

The data model is derived strictly from the FDA branded food database documents. There is one “class” defined for each document structure, and this is represented in the node.js project as schema definitions used by Mongoose.



Wireframes (UI)












The UI/UX design was started from wireframes that were created to meet the needs of the use cases. These wireframes are just mockups of the general idea behind the user interactions.




















The wireframes defined the ability to filter the products returned for a nutrient search by the minimum and maximum values in the specified units. This was not implemented in phase 1 but will be added in phase 2.

Structure

The project is structured using a hierarchical organization. This structure is defined as follows:

 nutribyte	/models contains the schema definitions .
 .git	/modules contains common server-side functions.
 models	/public contains all public resources used by the browser (client-side).
 modules	/public/css contains the cascading style sheet.
 node_modules	/public/images contains the images used on the web page.
 public	/public/scripts contains the scripts downloaded to the browser and used to define all dynamic behavior in the browser.
 css	/routes contains the definition of all server-side APIs used by the client-side javascript functions.
 images	/views contains all HTML files used to create the user interface presentation.
 scripts	
 routes	
 views	

The root of the project contains all directories, as well as the README.md file, server javascript file, package.json, and the design model and documentation. This is shown in the following image:

Name	
 .git	
 models	
 modules	
 node_modules	
 public	
 routes	
 views	
 .env	
 .gitignore	
 ACS5423 Project Report - Phase 1.docx	ACS5423 Project Report – Phase 1: this report document.
 ACS5423 Project Report - Phase 1.pdf	NutriByte Design Report: The design documentation produced by the Enterprise Architect product from the model.
 App.js	NutriByte Design.eapx : The Enterprise Architect design model itself. This file can only be processed by the Enterprise Architect product.
 NutriByte Design Report.docx	package.json: the project package file defining the required modules and structure of the node.js project.
 NutriByte Design.eapx	README.md: The read me markdown file that describes the project and which will appear in the GitHub repository main page.
 package.json	App.js: the main entry point for the Node.js project.
 package-lock.json	
 README.md	

GitHub Repository

All source code and design artifacts are committed to a personal public repository on GitHub named **ACS5423**. The URL to access the repository is: <https://github.com/dewayneh57/ACS5423>

CI/CD Pipeline

The initial CI/CD pipeline was created using GitHub actions. Currently, the CI actions are performing a node.js test (currently not defined). This will run test cases which are yet to be developed. The CI pipeline is triggered on a push to any branch, and will build and run tests on the code on that branch.

The CD pipeline is currently a no-op, and is triggered on a pull-request being approved and merged to the main branch only. This pipeline will deploy the application to the cloud system in phase 2.