Introduction

Alpha-Gal Syndrome is a disease that is caused by the bite of a lone star tick. Once the tick has bitten, the victim will now have an allergy to the sugar galactose-1-3-galactose (alpha-gal). Development of alpha-gal syndrome following the bite of a lone star tick can lead to a range of symptoms including hives, GI distress, and life-threatening anaphylaxis from exposure to mammal products.

Like Alpha-Gal, food allergies and allergic reactions are a problem that many people face (roughly 50 million Americans experience yearly allergies) in their daily lives. Looking out for products that could aggravate these allergies can be very difficult for college students that are living on their own for the first time. Knowing what foods they can and cannot eat is essential. For example, anaphylactic shock can take up to a week to recover from for life-threatening anaphylaxis from exposure to mammal products.

Alpha-Gal Syndrome is a disease that is caused by the bite of a lone star tick. Once the tick has bitten, the victim will now have an allergy to the sugar galactose-1-3-galactose (alpha-gal). Development of alpha-gal syndrome following the bite of a lone star tick can lead to a range of symptoms including hives, GI distress, and life-threatening anaphylaxis from exposure to mammal products.

Like Alpha-Gal, food allergies and allergic reactions are a problem that many people face (roughly 50 million Americans experience yearly allergies) in their daily lives. Looking out for products that could aggravate these allergies can be very difficult for college students that are living on their own for the first time. Knowing what foods they can and cannot eat is essential. For example, anaphylactic shock can take up to a week to recover from for life-threatening anaphylaxis from exposure to mammal products.

Objective

To solve this problem, this project will implement an iOS application where students can easily view allergen information for on-campus dining halls at Virginia Tech and barcoded food items from grocery stores. The main goal is to make the system faster and easier to use than the current Virginia Tech Dining Services website, to alleviate the stress associated with eating with allergies for students and residents of Virginia Tech/Blacksburg. The goal is to make the system faster and easier to use than the current Virginia Tech Dining Services website, to alleviate the stress associated with eating with allergies for students and residents of Virginia Tech/Blacksburg.

Application Implementation

As the main focus of this application is to provide easier access to allergen and nutritional information for Virginia Tech students, one of the themes we chose to highlight in this project is accessibility as a whole. Apple equips its devices with a setting available to every user to enlarge text sizes as a means to facilitate reading for users with poor vision. We intend to use each user’s device settings to determine the size of images and text that appear throughout the application. As a stretch goal, we also intend to employ text-to-speech technology to accommodate those with dyslexia. This focus on accessibility will ensure that all groups of people will have a streamlined experience while using our application.

The application is made up of three main tab bar views: Restaurants List, Barcode Scanning, and Settings. Each of these functions has its own corresponding sub-views that provide the user with everything they need to make informed eating decisions.

The second tab bar view is the barcode scanner, which is used to give information about barcoded food items found at grocery stores. This functionality is powered by the Nutritionix API, which receives a barcode and returns relevant nutritional information about the food item. The user will be shown an error message if no data is available for a given barcode, the user is shown an error message.

The third tab bar view is the settings tab. Here, the user can specify which allergies they have allergies or sensitivities to, and the data displayed to them will be filtered to give them only the information they need.

Restaurant List Design

The primary view for this application is the restaurants list. This view is composed of a list of the supported restaurants found on campus with their corresponding risk levels for each allergen. This risk level is determined by the percentage of dishes offered at the restaurant that contain the particular allergen. This data is collected when the Virginia Tech Dining Services website is scraped.

To represent the data, color-coded icons associated with each major allergen will be used to convey the risk level for each allergen. As allergic reactions can occur even when no ingredient contains the allergen in question, additional soft data will be collected and shared by users in the form of posts that are associated with each restaurant. Users will have the ability to upvote posts, and posts will be sorted in order of rating to ensure the most useful posts are displayed first. These posts will give users access to data concerning cross contamination, mislabeled dishes, staff accommodation, etc.

Clicking on a restaurant takes the user to the restaurant’s details page. This view displays an image of the restaurant, along with the restaurant’s online allergen menu, if available. Below that, the restaurant’s dish list is displayed by default, which is shown in Figure 2. Here the user will find the dishes currently being served at the particular restaurant. As in the restaurants list, each dish will have a risk level for each allergen, as well as a description of the dish. Clicking on a dish from a restaurant’s dish list will redirect the user to a full breakdown of the dish’s ingredients and relevant nutritional information about the dish. Using the segmented control on the restaurant’s details view, the user can switch to the list of posts for the restaurant, as shown in Figure 3. Each user has the ability to make a post related to the allergies at the restaurant in order to help other users. Furthermore, each post can be upvoted to denote usefulness. Posts are limited to 25 characters for the title and 250 characters for the body.

How We Obtain The Data

Since there is not a centralized data source that we can use to acquire information about Virginia Tech Dining Services, we are scraping and storing the data ourselves using Google’s Firebase. The application should pull accurate data from the scraping website for each individual item and store it on a weekly basis. Furthermore, it should be able to update itself and always have accurate data.

The workaround we implemented involved using Firebase as our main backend where we will store the data ourselves and collect it by scraping it with cloud Firebase functions. The current technique we are using to scrape the data from kosherdine.com is a headless JavaScript code that utilizes the library ‘puppeteer’ to scrape the data from each item and store them in a Cloud Firestore instance in the format shown in Figure 5. This script can be configured to run daily or weekly. As of now the script runs weekly but may be changed to run daily since available items can vary from day to day. The script provides fresh data for all users to access.