CS 4624
Multimedia, Hypertext, and Information Access

Final Report
11 May 2022

SafeEats
Virginia Tech, Blacksburg, VA 24061

Instructor: Prof. Edward Fox
Client: Two Alpha Gals
Team: Kevin Choo, Vlad Munteanu, Pranav Chavvakula
# Table of Contents

Table of Figures .................................................................................................................. 2

Table of Tables ...................................................................................................................... 3

Introduction .......................................................................................................................... 5
  Background ......................................................................................................................... 5
  Objective ............................................................................................................................ 5
  Deliverables ....................................................................................................................... 5
  Client ................................................................................................................................. 6
  Team ................................................................................................................................ 6

Main Issues With Current System ...................................................................................... 7

Requirements ....................................................................................................................... 8
  Data Collection .................................................................................................................. 8
  Data Representation ......................................................................................................... 9
  Barcode Scanner API ....................................................................................................... 9

Design ................................................................................................................................ 9
  The Original Restaurant List Design ............................................................................ 10
  The Latest Safe Eats Design ......................................................................................... 10

Implementation .................................................................................................................. 13
  Data Scraper Implementation ....................................................................................... 13
  Application Implementation ......................................................................................... 13
  Our Improved Stack ...................................................................................................... 15

Testing ................................................................................................................................ 15
  Manual Testing ............................................................................................................. 15
  User Testing .................................................................................................................. 15

User Manual ....................................................................................................................... 16
  Use Environment .......................................................................................................... 16
  Use Cases ...................................................................................................................... 16

Developer’s Manual .......................................................................................................... 17
  Issues with Project ....................................................................................................... 24
  VTURCS Marston Entrepreneurial Award (3rd Place) ............................................... 24
  Continuation ............................................................................................................... 25

References .......................................................................................................................... 26
Table of Figures

Figure 1: Screenshot of Server Technology being used by Virginia Tech Dining Services .............7
Figure 2: Screenshot of Current Dining Hall Lookup ........................................................................8
Figure 3: Original Wireframe ...........................................................................................................9
Figure 4: Restaurants List View .......................................................................................................10
Figure 5: Restaurant Dish List .........................................................................................................10
Figure 6: Restaurant Posts List .........................................................................................................10
Figure 7: Dining Halls Home View ..................................................................................................11
Figure 8: Restaurant Allergens .........................................................................................................11
Figure 9: Restaurant Dish List .........................................................................................................11
Figure 10: Restaurant Post List .........................................................................................................12
Figure 11: Global Search Food Items ...............................................................................................12
Figure 12: Settings Tab .....................................................................................................................12
Figure 13: JSON structure .................................................................................................................13
Figure 14: Determining Restaurant Safety .........................................................................................16
Figure 15: XCode on the App Store on Mac .....................................................................................17
Figure 16: XCode Developer Website ...............................................................................................18
Figure 17: GitHub Repository Where our Files Reside ................................................................19

   https://github.com/kchoo99/Safe-Eats/tree/main

Figure 18: Cloning project into a local directory of your choosing .........................................................20
Figure 19: Opening the project in XCode ..........................................................................................20
Figure 20: XCode landing page .........................................................................................................21
Figure 21: XCode repository page ....................................................................................................22
Figure 22: XCode branch selection for project ..................................................................................22
Figure 23: XCode cloning project into selected folder ......................................................................22
Figure 24: VTURCS Martson Entrepreneurial 3rd Place Award ..........................................................25
Figure 25: GitHub Repository ..........................................................................................................25
Table of Tables

Table 1: List of Files........................................................................................................................................23
Abstract

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 who 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, in severe cases. With the busy schedules and newfound independence associated with being a college student, it is important that access to information concerning allergens is easily available.

To solve this problem, we implemented an iOS application where students can easily view allergen information for on-campus dining halls at Virginia Tech. The main goal was to make the system faster and easier to use than the current Virginia Tech Dining Services website, in order to provide students with an easy way to check allergen information on the go. As it stands, Virginia Tech’s current implementation of distributing nutritional information about the food they serve is quite outdated, relying on a Windows IIS/8.5 server, with search latencies reaching upwards of 8 seconds and over 6 seconds for lookup latency. So, we worked to streamline the presentation of Virginia Tech Dining Halls nutritional and allergen data for students and faculty with allergen risks at Virginia Tech.

The final deliverables of this project include a partially functioning iOS application, a final report, and a set of presentation slides submitted to our sponsor, Dr. Edward Fox, our clients, Deborah Nichols and Candice Matthis, as well as Ashley Foster from Virginia Tech Dining Services. The report and presentation have progressed through our stages of development, including the project initiation -- in which we received criteria from the client and broke down deliverables, the planning stage -- in which a list of tasks was compiled, the execution stage -- which made up most of the development process, the testing phase -- in which we deployed the application to a small sample group, and the closing process -- where the product was presented to the clients and sponsor.
Introduction

Background

Roughly 4% of adults in the United States have food allergies, and with 30,598 students, this equates to around 1,224 students at Virginia Tech that have food allergies of some sort [1]. These allergies can range from mild to severe, so it can be difficult for students with allergies to determine whether or not a meal is safe for them to eat. In addition to the seven major allergens, many suffer from an allergy to red meat. One version of this allergy is known as alpha-gal, which is caused by bites from a lone star tick [2]. This condition causes severe allergic reactions to red meat products. This can be a difficult allergy to manage, as many food items that are labeled as safe by the FDA can use red meat products in their production. One such example of this is white sugar, which some manufacturers produce using compounds derived from cows.

Objective

The primary objective of this project is to make allergen information for on-campus foods more easily accessible to Virginia Tech students suffering from food allergies and sensitivities. Currently, allergen information for dining halls is located on the Virginia Tech Dining Services website, requiring several navigation steps to view the data [4]. This project aims to simplify the ease of access by allocating the same data centrally into a convenient iOS application. We intend to create an application that is centered around a User-Centric Design, rather than the current slow and inefficient system now in place at Virginia Tech.

Deliverables

This project will provide Virginia Tech students who have allergies, with a means of verifying the safety of current on-campus food items. To achieve this goal, the following deliverables must be met:

1. A data collection system that will scrape data from the Virginia Tech Dining Services website to obtain current dining hall nutrition information.
2. A representation of risk levels for each individual dining hall restaurant for students to determine the overall safety of a restaurant.
3. A representation of dish data from each individual dining hall restaurant for students to determine whether or not a particular dish is safe to eat.
4. A search tab for students to have access to a centralized database so they can quickly obtain information about their desired meals.
Client

The client for this project is the Two Alpha Gals, specifically Deborah Nichols and Candice Matthias. The Two Alpha Gals are an organization created to spread awareness about alpha gal allergies. Both clients provide valuable insight on the alpha gal allergy, as well as additional resources for allergies in general. Their goal is to spread awareness about Alpha-Gal Syndrome and we intend to accomplish this by clearly stating the risks of Alpha-Gal Syndrome within our application.

This project will eventually be delivered to Ashley Foster from Virginia Tech Dining Services, and deployed on the iOS App Store for Virginia Tech students to use. Ashley Foster is responsible for the dissemination of allergen related information and will be able to promote the application to students.

Deborah Nichols and Candice Matthias
Two Alpha Gals
thegals@twoalphagals.com
https://www.twoalphagals.com

Ashley Foster
Virginia Tech Dining Services
asha1089@vt.edu
540-231-1896
144 New Hall West, Blacksburg, VA 24061

Team

Our team is composed of Kevin Choo, Vlad Munteanu, and Pranav Chavvakula. All members are senior computer science students at Virginia Tech that have backgrounds in iOS application development.
Main Issues With Current System

The current state of the lookup for nutrition and allergen data from Virginia Tech is quite lackluster. As it stands, there is a search latency of 8.5 seconds and lookup latency of 6.3 seconds. A major reason for these long delays in search and lookup can be attributed to an outdated server. The system runs Windows IIS 8.5, which was released roughly 9 years ago. On top of the data being hard to fetch, some Dining Halls have specific websites that students will have to work through to obtain the data they need, further complicating the process of obtaining high quality allergen and nutritional information. Figure 1 shows a screenshot of the Server Technology being used by http://foodpro.dsa.vt.edu/menus/.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>private</td>
</tr>
<tr>
<td>Content-Type</td>
<td>text/html; charset=utf-8</td>
</tr>
<tr>
<td>Location</td>
<td><a href="https://foodpro.students.vt.edu/menus/">https://foodpro.students.vt.edu/menus/</a></td>
</tr>
<tr>
<td>Server</td>
<td>Microsoft-IIS/8.5</td>
</tr>
<tr>
<td>X-AspNet-Version</td>
<td>4.0.30319</td>
</tr>
<tr>
<td>X-Powered-By</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>Strict-Transport-Security</td>
<td>max-age=15768000; includeSubDomains; preload</td>
</tr>
<tr>
<td>X-XSS-Protection</td>
<td>1;mode=block</td>
</tr>
<tr>
<td>Date</td>
<td>Thu, 28 Apr 2022 00:24:55 GMT</td>
</tr>
<tr>
<td>Content-Length</td>
<td>155</td>
</tr>
</tbody>
</table>

Figure 1: Screenshot of Server Technology being used by http://foodpro.dsa.vt.edu/menus/

Figure 2 shows the Dining Services menu loading a request slowly.
Requirements

We needed to fulfill several requirements in order to satisfactorily complete the project. These requirements are detailed in the subsections below.

Data Collection

The core requirement for this application is that it scrapes and holds data for allergens contained in each meal that dining halls offer at Virginia Tech. Since there is not a centralized data source that we can tap into, we are scraping and storing the data ourselves using Google’s Firebase [5]. The application should pull accurate data from the dining hall 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.
Data Representation

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.

Barcode Scanner API

We originally wanted to provide the ability to scan grocery store food items by using the Nutritionix API. This API provides all allergen data and nutrition information for the majority of food items [6]. However, the cost of use per month for the API was simply too high to be worth implementing this semester. Future work can be done with this by finding an alternative to the Nutritionix API or perhaps a team will be able to implement their own version of a Barcode Scanner.

Design

Figure 3 shows the original wireframe for this project. It provides functionality for users to view various restaurants with their allergen risk levels, but omits the list of dishes served at each restaurant. Furthermore, it gives the user the ability to add additional restaurants to the list of restaurants displayed to all users to account for off-campus restaurants. This feature was ruled out to prevent scope creep, as restaurants would need to be validated before being shown to users.
The Original Restaurant List Design

The primary view for this application was originally 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. Now this page resides on the home page.

The Latest Safe Eats Design

As seen in Figures 7-12, these screenshots are significantly different from what we began with, as shown in Figures 4-6. A major drawback of the current VT Dining Services website is the number of navigations required to perform a query for a dish served at a particular restaurant. After considering what we wanted to accomplish with this project as a whole, we decided to alter our UI to match our goal of making queries as fast as possible following a user-centric design. The new homepage is now a list of dining halls at Virginia Tech that are displayed in a gallery view. We found that without sorting the numerous on-campus restaurants at Virginia Tech, the list of restaurants was very cluttered, and it was difficult to find the restaurant you were looking for. Clicking on any dining hall brings up a view that displays the restaurants located in that dining hall. This view is similar to previous versions of our application, just relocated to a new space.

Unfortunately, we had to scrap the barcode scanner portion of our application, due to Nutritionix API user limitations. We have replaced this tab with a global dish search for users to more
quickly determine the contents of a particular dish. The final tab, settings, has remained unchanged, with users having the ability to select their allergen sensitivities to declutter the UI somewhat and further speed up their queries.

Figure 7: Dining Halls Home View
Figure 8: Restaurant Allergens
Figure 9: Restaurant Dish List
Figure 10: Restaurant Post List
Figure 11: Global Search Food Items
Figure 12: Settings Tab
Implementation

Data Scraper Implementation

As stated previously, there is no centralized data source for dining hall data at Virginia Tech that we can use at our disposal. We attempted to reverse engineer the back-end for foodpro.ds.vt.edu, but the webpages are generated server side so there isn’t an API for us to tap into.

Thus, the solution that we came up with is two-fold. We are using Firebase as our main back-end 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 foodpro.ds.vt.edu is having JavaScript code that utilizes the library “puppeteer” to scrape the data for each item and store them in a Cloud Firestore instance in the format shown in Figure 13. 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.

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: Dining Halls List, Dish Search, 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 most useful view within our application is Dining Halls List, which is shown in Figure 7. Here, the user will find a list of every on-campus dining hall at Virginia Tech. Clicking on any dining hall will display a list of restaurants located in that dining hall, and users will know at a glance how safe the restaurant is for them to eat at. These restaurant risk levels provide valuable information to users about cross contamination and overall likelihood of having a reaction to their food.

Clicking on a particular 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 9. 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 details view, the user can switch the view to the list of posts for the restaurant, as shown in Figure 10. Any user has the ability to make a post related to the allergens 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. The only restaurant that remains unimplemented in the current version of the application is Owens Food Court, as employees insisted we get approval from Virginia Tech Student Services before taking pictures of the restaurants within the dining hall.

The second tab bar view was supposed to be 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 the information about the particular food item. However, due to the monthly cost of using the API itself, we decided against implementing it since only 1-2 users would be able to make use of this feature should we have used the free option. The concept went as follows: if an item is successfully scanned, the user will be redirected to a page displaying the ingredients and nutrition information for the item. If no data is available for a given barcode, the user is shown an error message. It is definitely plausible to turn the Barcode Scanner functionality into a future step for the upcoming semesters.
This tab has since been replaced with a global dish search tab, where users can search through all dishes served at Virginia Tech. This tab allows users to look for foods that they want to eat, but lets them know which options are safe.

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

**Our Improved Stack**

As mentioned earlier, the current state of the Virginia Tech Dining Hall nutritional lookup is quite poor in terms of user experience and latency. Our improved system aims to take advantage of Google Cloud Services to be able to significantly decrease latency time and make substantial improvements to the user experience. This was accomplished by having our data completely centralized, meaning that no matter if the Dining Hall is part of a larger chain of restaurants (i.e., Chick Fil A), the data will be stored in one centralized server.

**Testing**

**Manual Testing**

One team member, Kevin Choo, has a severe peanut allergy, and has tested the app regularly to ensure practicality of the application. Furthermore, we verified the data we scraped with Ashley Foster from Virginia Tech Dining Services, as mislabeled data could be life threatening for potential users of the application.

**User Testing**

A group of 15 Virginia Tech students with food allergies was assembled to test the application. The testing period was a week long, and its results gave us insight into which areas of the application needed refinement.

**User Feedback**

Following user testing, a survey was sent out to testers. The current version of the application was built upon dummy data, and this round of testing was conducted to address UI problems and inconveniences. Most responses were associated with known bugs in our system, including many UI bugs.
User Manual

Use Environment

This application is meant to be used by Virginia Tech students with allergies. Dining hall information will only be supplied for on-campus restaurants at Virginia Tech.

Use Cases

Determine the safety of a dish
To find information about a specific dish, navigate to the global dish search tab. Type in the search criteria, i.e., the name of the dish. Click on the desired dish, and review the information.

Determine whether a restaurant is safe to eat at
To determine the risk of a particular restaurant, locate the dining hall of the restaurant you wish to dine at, and click on it. You will be brought to a list of restaurants located in the dining hall. Locate the restaurant you wish to dine at, and note the allergen risk icons for that restaurant. If additional information is needed, click on the restaurant and you will be directed to a list of dishes served at the restaurant. You can also switch to the Post List tab using the segmented control to gain additional information.

Figure 14: Determining Restaurant Safety
Developer’s Manual

Prerequisites
Prior to working on this project, developers will need to have a MacOS machine (or simulator, i.e., MacOS VM). Developers will also need to install the XCode Development Studio onto their compatible MacOS Machine. This can be found through the App Store on Mac (as shown in Figure 15).

![Figure 15: XCode on the App Store on Mac.](image)

Alternatively, developers can go to the Apple Developer website (as shown in Figure 16) [https://developer.apple.com/xcode/](https://developer.apple.com/xcode/).
Cloning the Project to your Local Machine (Method 1)

Once XCode has been installed onto your machine, it is time to pull the files from GitHub to the local machine. Navigate to the link listed in Figure 17 to pull up the Safe Eats Repository. Here, we must obtain the HTTPS address shown when the green ‘Code’ button is clicked. The link is also listed here.  https://github.com/kchoo99/Safe-Eats.git
Now that we have obtained the repository URL, determine a location on the local machine where the code will reside. In the example listed in Figure 18, the location where the files will reside will be in the 4624 folder located within the desktop of the local machine. In a terminal, navigate to the folder where the files are to be placed and run `git clone https://github.com/kchoo99/Safe-Eats.git`. If this step is completed correctly, developers will see something similar to what is shown in Figure 18.
Once the files have been cloned into the appropriate folder on the local machine, we can boot up XCode to open the project. Once XCode is launched, in the task bar, click on File->Open. This will now open a Finder menu for you to select an XCode project to Launch. Navigate through the Finder menu until the Safe-Eats directory is opened. Click on the Safe Eats.xcodeproj icon (the item with the App Store Logo), then press the blue ‘Open’ button to open it in the XCode development environment.
Once you click ‘Open’, a screen similar to the one shown in Figure 20 will appear in the XCode Development environment.

Cloning the Project to your Local Machine (Method 2)

Prerequisites

On top of the aforementioned prerequisites, this method requires that the developer link their GitHub account to XCode itself. Instructions on how to do so can be found at the following Apple Developer Page:


In the XCode taskbar, select Source Control->Clone. This will bring up a list of repositories connected to your GitHub account. Select the ‘Safe-Eats’ repository and then click the blue ‘Clone’ button in the bottom right hand side, as shown in Figure 21. Once ‘Clone’ has been clicked, select the ‘main’ branch and click the blue ‘Clone’ button once again.

Similarly to how a Finder window was opened, as shown previously in Figure 20, navigate to the desired folder on the machine. Hit ‘Clone’ again, and a landing screen similar to Figure 21 will appear.
Now developers may begin their development on the application with Source Control. We recommend that each developer has their own branch to avoid conflicts in the long run.
Files
We utilized a number of .swift files in the development of the application. BlurView.swift is a generic class that implements the gallery view featured on the homepage of Safe Eats. Home.swift utilizes BlurView.swift and the dining hall list to display a list of the dining halls available at Virginia Tech. DiningHallRestaurants.swift implements the restaurants list functionality, and uses a filtered list of all the restaurants at Virginia Tech to display the restaurants located within a particular dining hall. This view is displayed on a swipe-up sheet, so the user can easily navigate between dining halls. RestaurantDishes.swift defines the view used to display both the dishes and posts related to a particular restaurant. VoteButton.swift is the structure of the rating system used for restaurant posts. Search.swift is associated with the second tab bar view of Safe Eats, the global dish search. Settings.swift is the final tab bar view, and allows users to select their specific food sensitivities. SafeEatsApp.swift is the main class of our application, and it calls ContentView.swift to display all the views of the application. These files are summarized in Table 1.

<table>
<thead>
<tr>
<th>File Name</th>
<th>File Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlurView.swift</td>
<td>Basis of the home page</td>
</tr>
<tr>
<td>Home.swift</td>
<td>Builds the home page with the dining hall list</td>
</tr>
<tr>
<td>DiningHallRestaurants.swift</td>
<td>Pop Up sheet that lists the restaurants located within a dining hall</td>
</tr>
<tr>
<td>RestaurantDishes.swift</td>
<td>View that lists the dishes served at a restaurant</td>
</tr>
<tr>
<td>Search.swift</td>
<td>View that allows for global dish search capabilities</td>
</tr>
<tr>
<td>Settings.swift</td>
<td>View that allows users to select specific allergens</td>
</tr>
<tr>
<td>VoteButton.swift</td>
<td>Sub View that allows users to vote on posts</td>
</tr>
<tr>
<td>SafeEatsApp.swift</td>
<td>Calls the Content View</td>
</tr>
<tr>
<td>ContentView.swift</td>
<td>Establishes the Tab Bar View and renders the home page</td>
</tr>
</tbody>
</table>

Table 1: List of Files

Setting Up Scraper Back-end
Our scraper utilizes a Javascript script that uses the library “cheerio” to parse through the foodpro.dsa.vt.edu webpage for each restaurant. Currently, we use our script as a Firebase cloud function which updates a Firebase realtime database. This is fairly easy to set up by cloning our
code in the repository and setting up your own Firebase project with a cloud function using our script.

Issues with Project

Barcode Scanner Implementation
One of the major issues that we had encountered for our implementation was the Barcode Scanner for foods off-campus such as grocery stores such as Food Lion and Kroger. Our biggest limitation in implementing the barcode scanner was the cost of using the API itself. We originally planned to use the Nutritionix API, but later discovered that the free version only supports a maximum of two active users. Expanding this functionality would cost a significant amount of money, which led us to scrap the idea.

Work to be Done
Due to the fact that the Owens Food Court employees would not allow us to photograph the restaurants located within the dining hall, none of these restaurants appear on the Restaurants List. Currently, a placeholder consisting of an Au Bon Pain restaurant item appears. This is to prevent image unavailable errors; other solutions could easily be implemented. Furthermore, the Post List is not currently connected to the back-end, and this could also be addressed easily. Finally, the Dish List is not fully implemented, but this is due to a lack of data. The scraper is partially working, but is not fully functional yet. The final step would be to implement GrubHub ordering capabilities, which was a stretch goal for this project. As it stands, the application is not ready to be deployed to the App Store. Should the aforementioned items be implemented in upcoming semesters work, the application would be ready to be deployed to the App Store. Potential future clients for this project could be representatives from VT Dining who would benefit from the utility of the application. Moving forward, a completed project would need to be sustained through the maintenance of accurate data regarding the current food being served at each restaurant.

VTURCS Marston Entrepreneurial Award (3rd Place)
On April 24, 2022, the team attended the Virginia Tech Undergraduate Research in Computer Science Spring Research Symposium. We are able to present our project to various students, faculty, judges, etc. and receive helpful feedback to improve our project. Upon the conclusion of all the presentations, we were awarded third place for the Marston Entrepreneurial Award. We were extremely pleased with the results of the Symposium and were able to obtain a lot of potential future directions for us to take.
One of the judges who represented the CIA particularly enjoyed our product and gave us some possible future directions we could take with this project. One of them being expanding our reach to other campuses to provide them with a similar experience. This could be accomplished by just having other campuses send their Dining Hall data in a JSON format we specify, and allowing them to make use of the front-end that we have designed.

Contamination

A major concern for this project was scalability. Future extensions of this project can focus on adding support for off-campus restaurants, as well as deploying/adapting this system for other organizations and universities. All required materials will be provided to students wishing to continue this project in the future in the GitHub repository shown in Figure 25 [7].

Figure 24: VTURCS Martson Entrepreneurial 3rd Place Award
References


