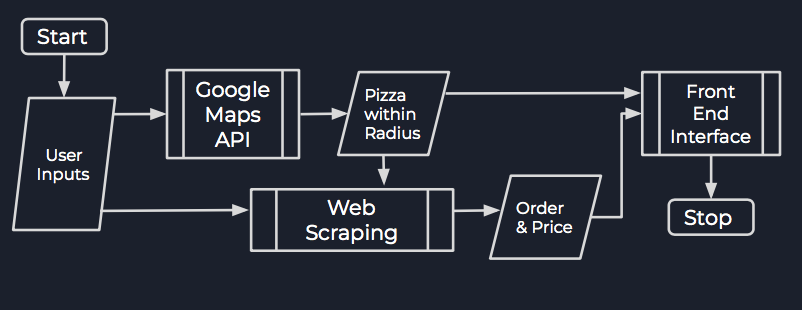
|  |
| --- |
| QEC Competition |
| Save that Dough |
| Programming Challenge – Team 1 |

|  |
| --- |
| Sam Alton, Raed Fayad, Laure Halabi, & Nathaniel Pauze  11-24-2018 |

# Overview

This report outlines how the “Save that Dough” front-end and back-end software will function. Primarily, the front-end software is locally hosted and will prompt the user for inputs in the order of: the current address of the user, a search radius (in kilometers), size of the pizza, and number of pizza toppings. This will iterate for however many pizzas the user requests. The data is stored into input fields of the site. The location address and the maximum radius size are sent to the back-end software that generates the list of pizzerias as well as their websites within the radius using the Google Maps API platform. The web scraping back-end software will access the dominos website. This process will search through the main page of the website for all the possible pizza orders based on the user’s inputs, whilst retrieving the prices for each and saving them into an array. Subsequently, the program will submit the final list of cheapest pizzas to the front-end platform to be displayed. The front-end platform will calculate the total cost of the order and display it on the screen.

Figure 1: Flowchart showcasing the overall functionality of the program

# Back-End Software

## Google Maps API

This program receives the input of the address as a string, formats it for the API by removing spaces and symbols and converts it into a URL for an API call for Google’s geocoding system. After receiving json a response, it finds the latitude and longitude for the address and formats it into a string. It then prepares and sends another google API call containing these geographic coordinates with the parsed input for search radius, along with hard coded parameters to find pizza locations. Finally, it iterates trough the response and outputs all the nearby pizza locations, saving them into a global array for the next step of the system. It can also conditionally only return locations that are open at the given time.

## Web Scraping

The web scraping software accesses the website for the pizzeria and goes through each element on the main page. Following this, the software cross references the information to the size of the pizza that the user’s inputted and looks for any matches. After a match has been found, the element on the website is stored into an array as well as the price of that order and the name of the pizzeria branch. The cheapest price is then sent to be displayed for the user.

# Testing

The team has conducted basic functionality testing that assures that all the features on the user interface platform function properly. However, the platform is assuming that the user will enter valid inputs. For the back-end software, the team used the unit testing technique to ensure that individual methods or classes are working as expected by testing across a range of valid inputs. In the future, the team would hope to include single-user performance testing which would test the system’s responsiveness and execution time and help the team improve the software’s performance.

# Limitations & Recommendations

To improve the current software the team has developed, the team would like to include a database that stores synonyms for the different sizes of pizza since users may enter different words that refer to one size. An example of this is using “Party Size” as a synonym for “Extra-Large” or “Single” for “Small”. Furthermore, the back-end software is currently only capable of accessing the data from the main page of some of the top pizzeria chains in Ontario. Given more time, the team would like to implement this program to work for all legitimate pizzerias and be able to access their full website. In addition, the software is not equipped to handle invalid inputs, so it would be best to implement that in the future.