1) An overview of the function of the code (i.e., what it does and what it can be used for).

The project goal is to search for a restaurant or recipe and display a list of results based off the query. When a result is clicked from the list on the results page, it will give more information on that result. This allows users to see information about recipes to possibly prepare food, or information about restaurants that may help the user determine if they'd like to eat there. More specifically the web application has a database where the data is stored from using zomato api (for restaurant information) bing api (for related images to the restaurants since zomato mostly did not have images for restaurants) and yummly api which was used for recipe database (files for api calls can be found in project/copy of important files/ zomato script or yummlyscript. Our application takes in a query depending on if the user is searching for a restaurant or recipe. The application will look at the query to look for matches from either typing in the exact name of the recipe/restaurant, a substring of the recipe name/ restaurant name, an ingredient name for recipes, cuisine type for recipes and restaurant (meaning if the query has specified a cuisine type then it will find all matching recipes or restaurant that have that particular cuisine type), and also if the user has put in a city name for the restaurant search query then the application will find all restaurants in the database with that particular location and return the results. Finally our most novel idea of the application is to provide the user a list of related recipes that are associated with a particular restaurant. When the user has selected a particular restaurant from his/her results or typed in the exact restaurant name then on the bottom of the page a link to recommended recipes which is a function that looks at the type of food served at the restaurant and then tries to find any recipes that have the same cuisine type as the restaurant to give the recommendation. If no matching cuisine can be found in the recipe database then a list of random recipes are presented by default to still provide some form of recommendation and this list of default recipes does change every time you refresh the page(default is only present if restaurant cuisine type has no matching recipe cuisine type in the recipe database) **(PLEASE PLEASE look at the documentation in the code as well which is very thorough on describing the search query functions which can be found in project/copy of important files/ views.py)**

2) Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

NOTE WALK THROUGH OF CODE IS PROVIDED IN THE VIDEO PRESENTATION STARTING AT 12:45

We used django framework to host our data and html code. Zomato and Yummly are the two API's that we are using to get our data. We used scripts to pull the data from the API's to store in our database.

project/project/database/views.py

(PLEASE PLEASE look at the documentation in the code as well which is very thorough on describing the search query functions which can be found in project/copy of important files/ views.py)

search_recipe and search_restaurant functions help set up the input form for the respective search webpages and connects the input to the search engine.

results_restaurants is the implementation of the restaurant search engine that takes the query input from the user and checks the database for related information.

- checks for the exact name for a restaurant and returns the information to the webpage if exact restaurant name is found.
- checks if it's a city, and returns the results if the query is a city (all restaurants in the particular city).
- checks to see if it's a cuisine type, and returns the information if a match is found (all restaurants with the user input cuisine type).
- checks if the query is a substring of a restaurant name and returns all matching restaurants that have the substring in their name.
- otherwise returns a no results found page.
- displays a link which directs a user to a new page with a list of recipes related to the food served at the restaurant.

results_recipe is the implementation of the recipe search engine that takes the query input from the user and checks the database for related information.

- Checks for exact name
- Checks for ingredients in a recipe
- Checks for cuisine type
- Otherwise returns a no results found page

A user can use the home page to search for recipes/restaurant. For example, from the homepage use the top navigation bar to go to restaurant search, type in a restaurant name, and a page of restaurants will come up. Click on a restaurant to see information on that restaurant.

3) Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

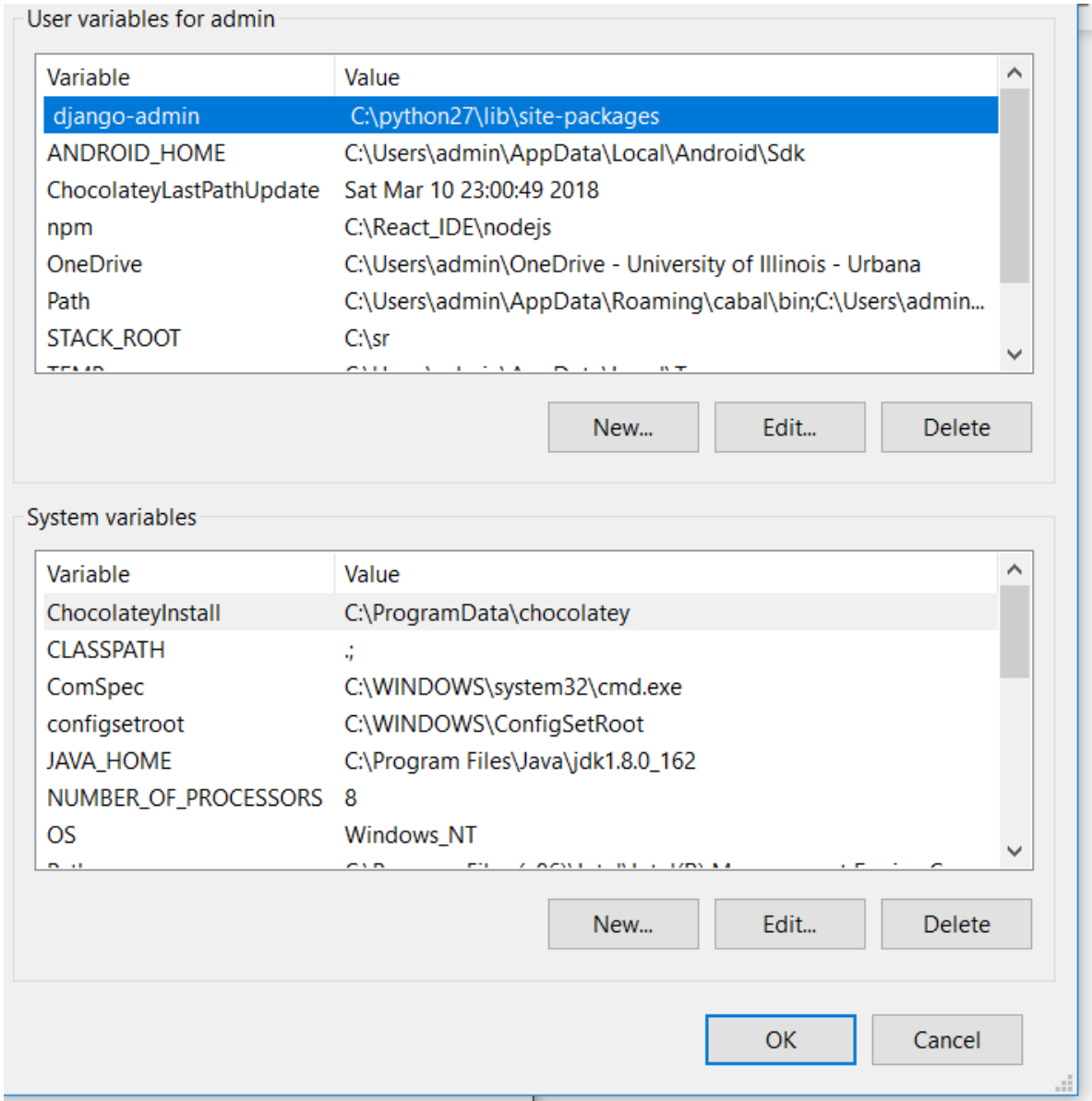
NOTE NOTE NOTE: tutorial on how to download and run software was done at the end of the video presentation

Install django. https://docs.djangoproject.com/en/2.0/topics/install/ (note on bottom right corner click version 1.11 since we used this version to create our project)

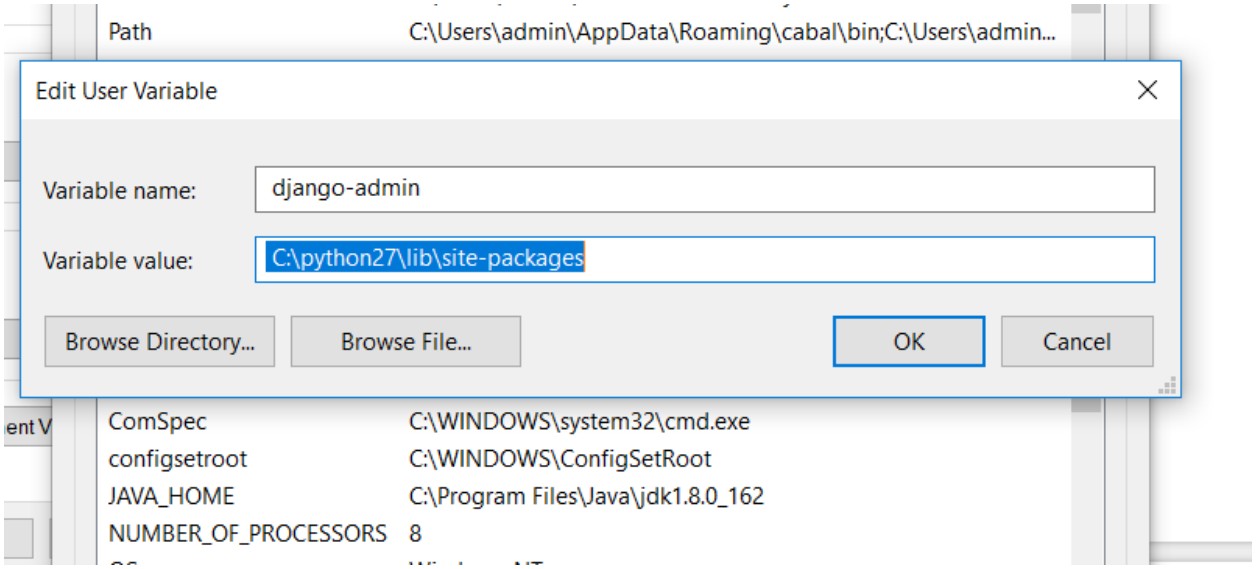
To install django enter the following prompt in (cmd or terminal) assuming you install pip package during your python installation

pip install -v django 1.11 (if you have python setup then you can use pip and please install python 2.7 that was the version used for this application)

You have to setup an environment variable which is called django-admin and the value/ path is rooted to your python library. So this is how it looks like for Windows.



ou have trouble setting up django or the path



Pull the project directory from github.

cd to the project/project. This directory should have the manage.py file make sure this is there before running this in terminal/cmd. Run `python manage.py runserver` locally. And on success you will get the following link 127.0.0.1:8000.

Navigate to `127.0.0.1:8000/home_page` in a web browser.

Please look at the end of our video presentation if you have trouble setting up django or the path variable to run the project

4) Brief description of contribution of each team member in case of a multi-person team.

Asad configured django and got the basic framework running. Implemented Search for recipe and Search for restaurants and results pages. Corrected issues with html pages. Implemented recipe recommendation function for a particular restaurant. He linked the html files as well as wrote scripts to get data from the API's for Zomato and configured Sally's script for Yummly.

Dykali worked on getting some basic search and result variables working to start displaying data from the back end framework on webpages. He designed the front end home page, search pages,

and results list and item pages. He also worked on getting the information to display correctly from search results.

Sally wrote a script to help populate the recipe database from the Yummly API.