Team eggsD -- Food for Thought

Abstract:

On the homepage, the user is prompted to input an ingredient or dish. There will be two types of searches. If the user decides to input text for their search, the user will be directed to a page with information about his or her search term. The user can also decide to input an image, which would load possible ingredients of the image. Then, upon clicking one of these ingredients, the user can access the information about that particular ingredient. The information page, for either of the searches, includes a wikipedia article, related images, recipes, and nutrition facts. The three APIs used are Clarifai, Wikipedia (MediaWiki), Yummly, and USDA Food Composition Databases. We have chosen to use bootstrap as our framework.

Optional:
Accounts: (would incorporate a database)
-Save ingredients
-Past searches

Database Schema:

Depending on the amount of time given, we might create a database for accounts so that ingredients can be saved and past searches are shown.
Potential database: username, password, saved ingredients, past searches

Timeline:

12/6:

- Jessica - create getIngredient() which returns a list of possible ingredients from an image (image recognition API)
- Issac - create getNutrition(), which returns nutritional information about an ingredient (nutritional facts API)
- Jerry - start working on app.py
- Richard - create home.html

12/7:

- Jessica - create getArticle() and getImage(), which return an article and related images respectively, about an ingredient (wikipedia search API)
- Richard - create list.html
- Issac - test getIngredient() with getNutrition()

- Jerry - finish the route for home.html on app.py

12/8:

- Jessica - test getIngredient() with getImage() and getArticle()
- Issac - create getRecipe(), which returns a recipe with the ingredient
- Richard - create info.html
- Jerry - finish route for list.html

12/9:

- Jerry - finish route for info.html
- Richard - work on css/framework (if personalization is necessary)
- Jessica and Issac - critique the templates and pathing

12/10:

- All members - beautify/make last suggestions

12/11:

- All members - debug

Task Delegation:

Richard Wang (Project Manager):

Specific tasks: list.html, info.html, home.html

Will be doing most/all of the frontend. Also, as project manager, manage this project (keeping people in track with the devlog and the timeline)

Jessica Yang:

Specific tasks: getArticle(), getImage(), getIngredient()

Will be using the Wikipedia API and Clarifai API to create these functions. Responsible for transferring the information to app.py. Also responsible for merging getIngredient() with her get functions.

Jerry Chang:

Specific tasks: app.py

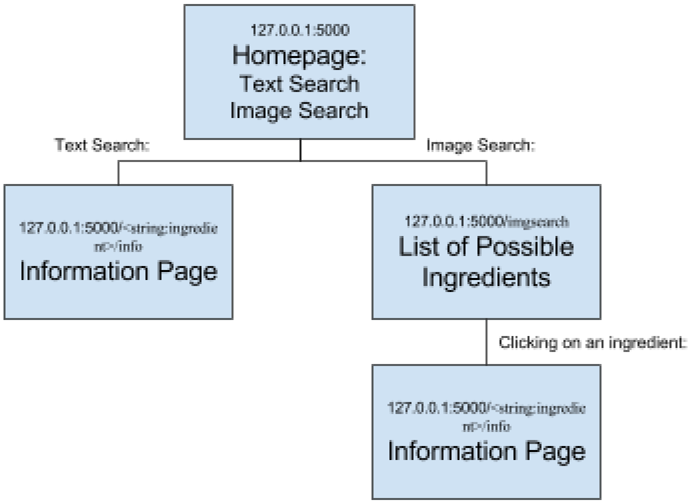
Responsible for creating the routes and sending the relevant information to the templates.

Issac Kim:

Specific tasks: getNutrition(), getRecipe()

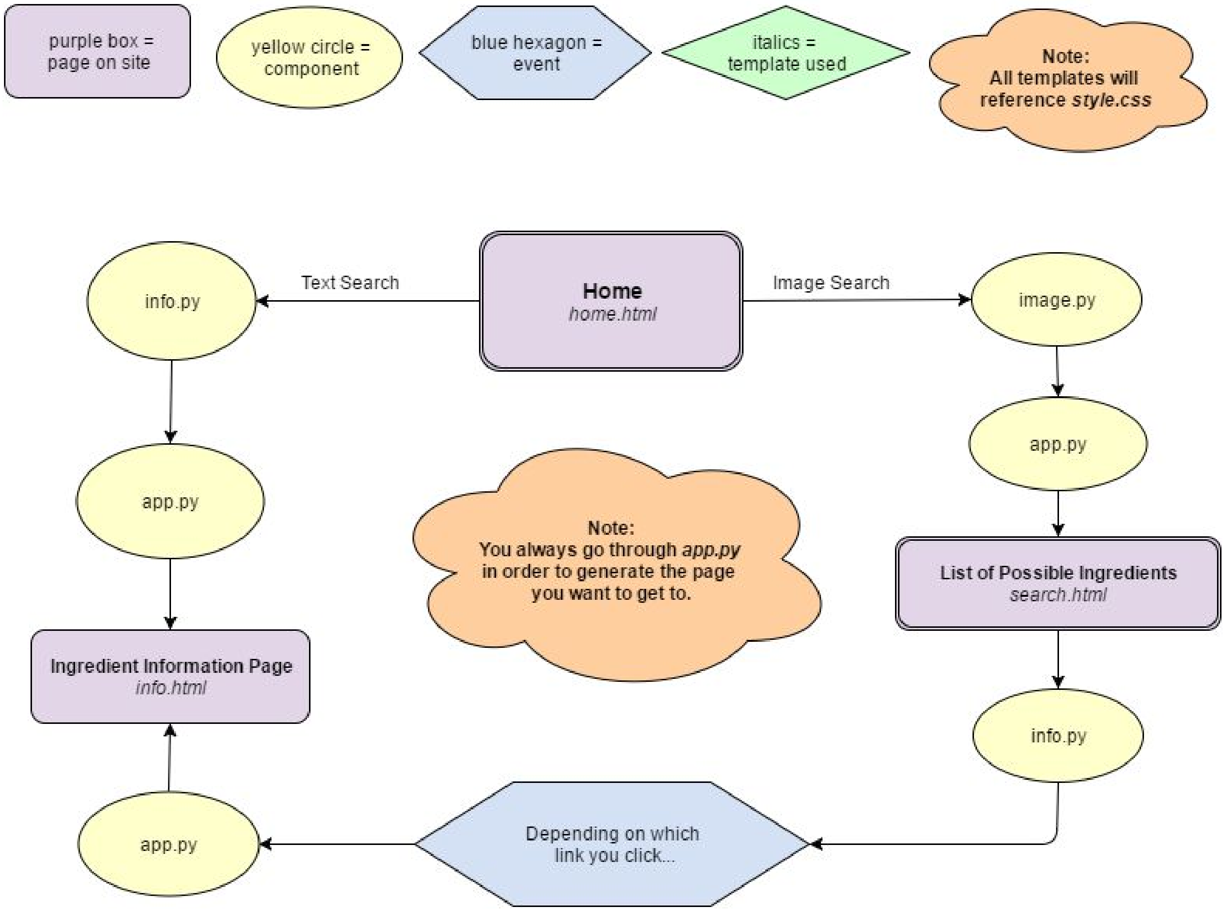
Will be using using the nutrition API and Yummly API to create these functions. Responsible for transferring the information to app.py. Responsible for merging getIngredient() with his get functions.

Site Map:


127.0.0.1:5000
Homepage:
Text Search
Image Search
Text Search:
Image Search:
127.0.0.1:5000/<string:ingredient>/info
Information Page
127.0.0.1:5000/imgsearch
List of Possible Ingredients
Clicking on an ingredient:
127.0.0.1:5000/<string:ingredient>/info
Information Page

Information page includes a wikipedia article, related images, recipes, and nutrition facts. There will be three templates: homepage template, possible ingredients template, information template.

Component Map:


purple box = page on site
yellow circle = component
blue hexagon = event
italics = template used
Note: All templates will reference *style.css*
Home *home.html*
Text Search
Image Search
info.py
image.py
app.py
app.py
Note: You always go through *app.py* in order to generate the page you want to get to.
List of Possible Ingredients *search.html*
Ingredient Information Page *info.html*
info.py
Depending on which link you click...
app.py

app.py
templates/home.html
templates/info.html
templates/list.html
utils/image.py
utils/info.py
static/style.css
static/css (bootstrap)

Component Description:

Templates (Bootstrap Framework):

- home.html
    - css
    - heading
    - text form:
        - Refers to route: “/<string:ingredient>/info”
        - ingredient is the name of the value
    - image form:
        - Refers to route: “/imgsearch”
        - imgsearch is the name of the value
- list.html
    - css
    - heading
    - list of links (for each possible ingredient):
        - Each link refers to “/<string:ingredient>/info”
- info.html
    - css
    - heading
    - wikipedia information
    - images
    - nutritional facts
    - recipes

API (Documentation):

Nutrition facts about foods:
https://ndb.nal.usda.gov/ndb/api/doc

Image recognition:
https://api.clarifai.com
Wikipedia API for articles and images:
https://www.mediawiki.org/wiki/API:Main_page#The_format
Recipe API:
https://developer.yummly.com/

Python:

- app.py
    - home() [route: “/”]
        - template: home.html
    - info(ing) [route: “/<string:ingredient>/info”] POST
        - template: info.html
        - ingredient: ing
        - messages: article (string), images (list), recipes (string), nutrition (dictionary)

          Description: The text input (“ingredient”) becomes the ingredient. getInfo() is called to retrieve information
    - possible() [route: “/imgsearch”]
        - template: list.html

          Description: An image (“imgsearch”) is uploaded. getIngredient() is called to return a list of possible ingredients, which can be clicked on to go to info(ing).

- image.py
    - getIngredient(url)
        - used if image is uploaded
        - displays a list of possible ingredients, which are linked to their information pages
    - handles image recognition
    - returns possible names of ingredients in an image

- info.py
    - getArticle()
        - takes in the text of an ingredient and returns a string consisting of wikipedia’s article
    - getImages()

-takes in the text of an ingredient and returns a list consisting of the urls of wikipedia’s images

-getNutrition()

-takes in the text of an ingredient and returns a dictionary of nutrition facts

-getRecipe()

-takes in the text of an ingredient and returns a string of the recipe

-handles information regarding ingredients