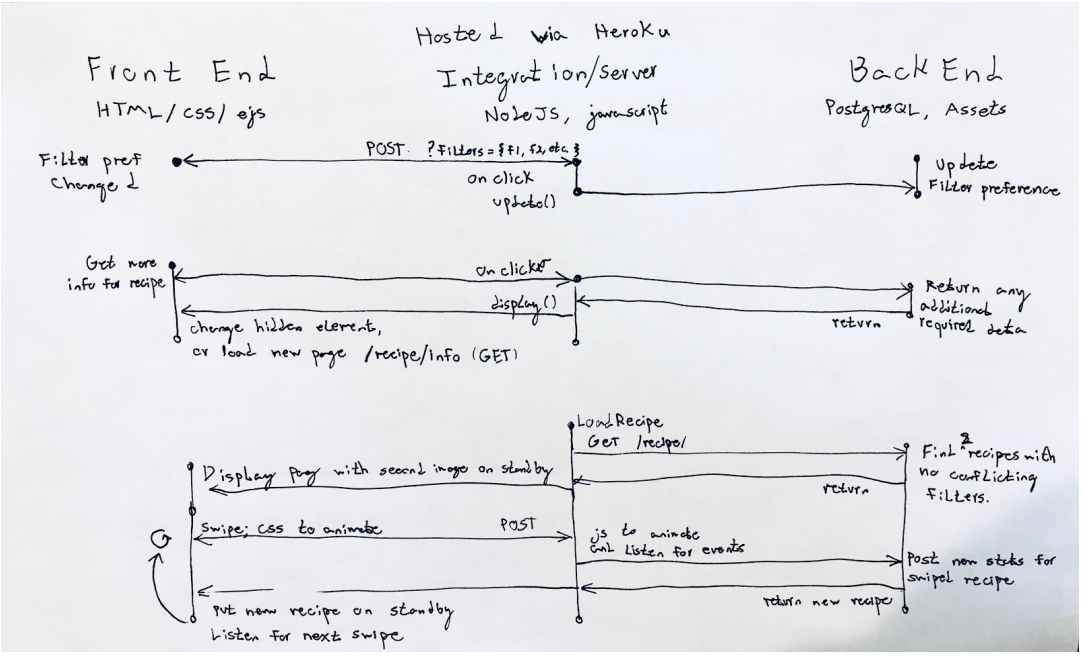
Team MergeSort(); Project Milestone 4

Revised List of Features:

Features	Functional	Non-Funcitonal
2) Filters	Restrict food options based on user input • 1) Vegetarian • 2) Vegan • 3) Gluten-Free • 4) Allergies • 5) Lactose-Intolerant • ~~Budget~~ DROPPED	Blacklists meals and only displays meals that fall under the categories chosen by the user
1) Swipe Left	User doesn't like this recipe	Tosses out these items so they will no longer be seen
1) Swipe Right	User does like this recipe	Saves the items that the user liked so they can reference them in the future
1) Swipe Up	Shows more details of the recipe	Displays a new page with the recipe and other potential information
3) Ranking	Based on the popularity of what users rated the recipe, based on stars out of 5	List is created (mix of highly ranked, random, and other attributes)
4) Reviews	Lists the comments that previous users that made this recipe had for it	Data stored about each recipe, full of strings of the different comments
~~Location~~ DROPPED	No longer necessary as we have pivoted to be a recipe app rather than restaurants on the hill	N/A

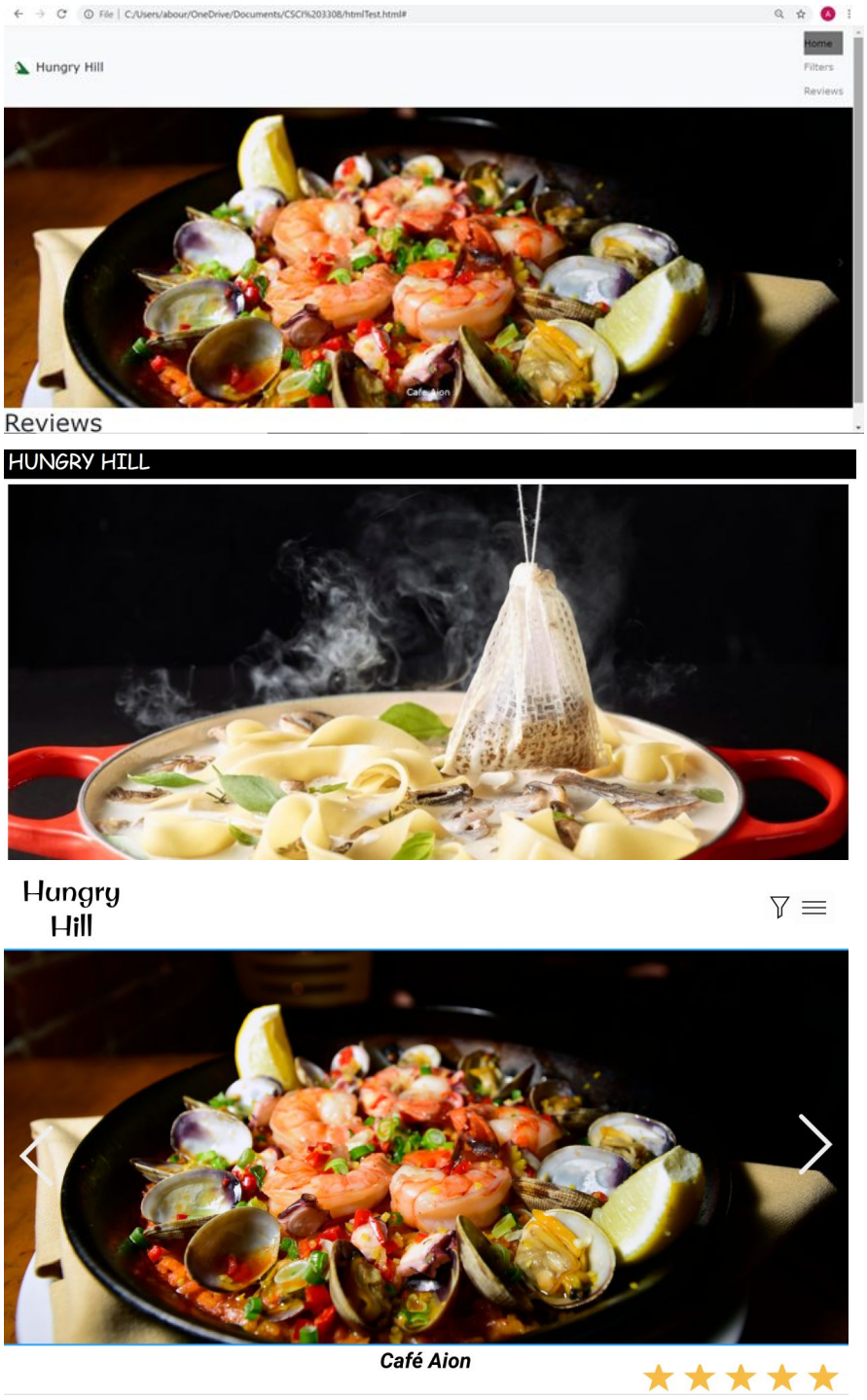
Architecture Diagram:

Layer Processes- Details user input watches on javascript, GET/ POST requests and handlers within server, and data return queries.



Software layer's relations- Details the fundamental interactions between layers and the purposes each interaction is meant to fulfill.

Front End Design:



We are currently working on doing some research with bootstrap, and have set a few styles of how our main page is going to look.

Web Service Design:

Hosting will be done over Heroku
API is currently to be determined

Database Design:

Summary design of application database; Explaining each table's purpose, and its respective records:

- RECIPE
    - Purpose of this table is to serve as the central relation amongst all the tables; that is, the table that is connected to all the information respective to each record (or recipe) in the database.
    - 'recipe_id' serves as the table's primary key, and thus is the unique identifier for each record/recipe in the table; stored as a integer datatype
    - 'name' stores the title of each recipe as a varchar datatype.
    - 'picture' stores an image of the recipe; datatype image.
- DETAILS
    - Purpose of this table is to store all text-based details that are relevant to each recipe.
    - 'detail_id' is the primary key of the table, and is the unique identifier for each recipe's set of details.
    - 'recipe_id' is the table's foreign key.
    - Remaining records in this table will all be varchar datatype.
- STATS
    - 'stat_id' primary key
    - 'recipe_id' foreign key
- DIETARY_ACCOMMODATIONS
    - 'Diet_id' primary key. This is the unique identifier for a recipe with certain dietary preferences/restrictions.
    - 'Recipe_id' is the foreign key.
    - The remaining records in this table will be boolean datatypes, that correspond to a certain dietary preference/restriction.

Datatypes being stored:

- Datatype: varchar
    - Records: description, prep_time, cook_time, directions, nutrition_facts, name, picture
- Datatype: Boolean
    - Records: vegan, dairy_free, vegetarian, gluten_free, nut_free
- Datatype: Integer
    - Records: detail_id, recipe_id, diet_it, stat_id, num_likes, num_dislikes

DBMS used:

- PostgresSQL

Database Schema (Entity Relationship diagram):