Writing 3 - Project Description

Zach Bloomstein, Max Englander, Don Kim, Sarah Stevens

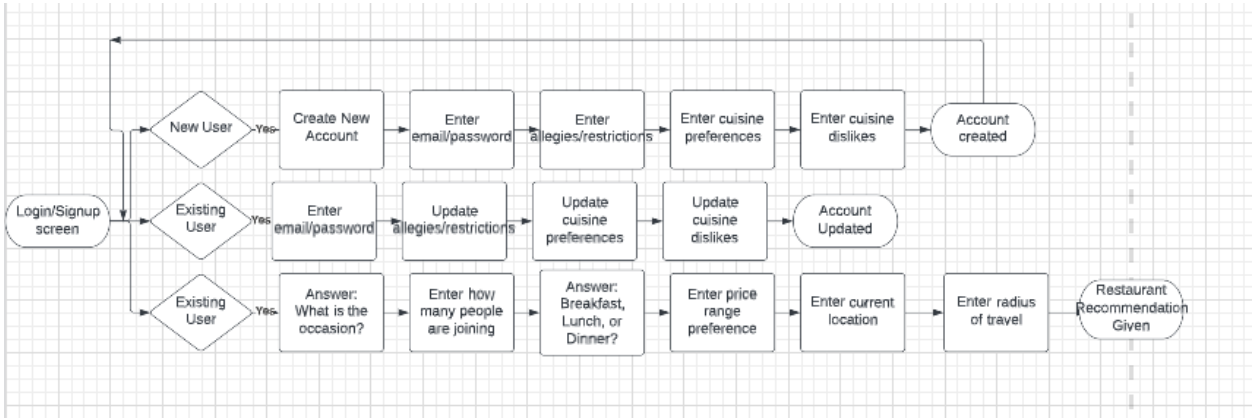
Product Specifications

User Stories

- As a new user, I would like to be able to create an account quickly so that I can get started on using the app as soon as possible.
- As a new user, I would like the application to be as intuitive and easy to use as possible so that I am not confused as to how to navigate it.
- As a new user, I would like my initial restaurant suggestions to be relevant for me so that I feel the application is worth my time.
- As a returning user, I would like to be able to alter my profile settings easily and quickly such as cuisine preferences or dietary restrictions so that future restaurant suggestions are more accurate.
- As a returning user, I would like to be able to update account settings such as my email address or password so that I can always have access to my account.
- As a returning user, I would like to see suggestions improve over time so that I feel that the application is becoming personalized to me.
- As a returning user, I would like for the questions that the application asks me everytime I use it to be relevant and important for the eventual restaurant I'd like to attend.
- As a returning user, I would like to get suggestions for restaurants I wouldn't normally think about going to so that I can have new experiences.
- As a user in general, I would like for the suggested restaurants to fit my dietary restrictions and not offer restaurants that are incompatible with my diet or lifestyle.

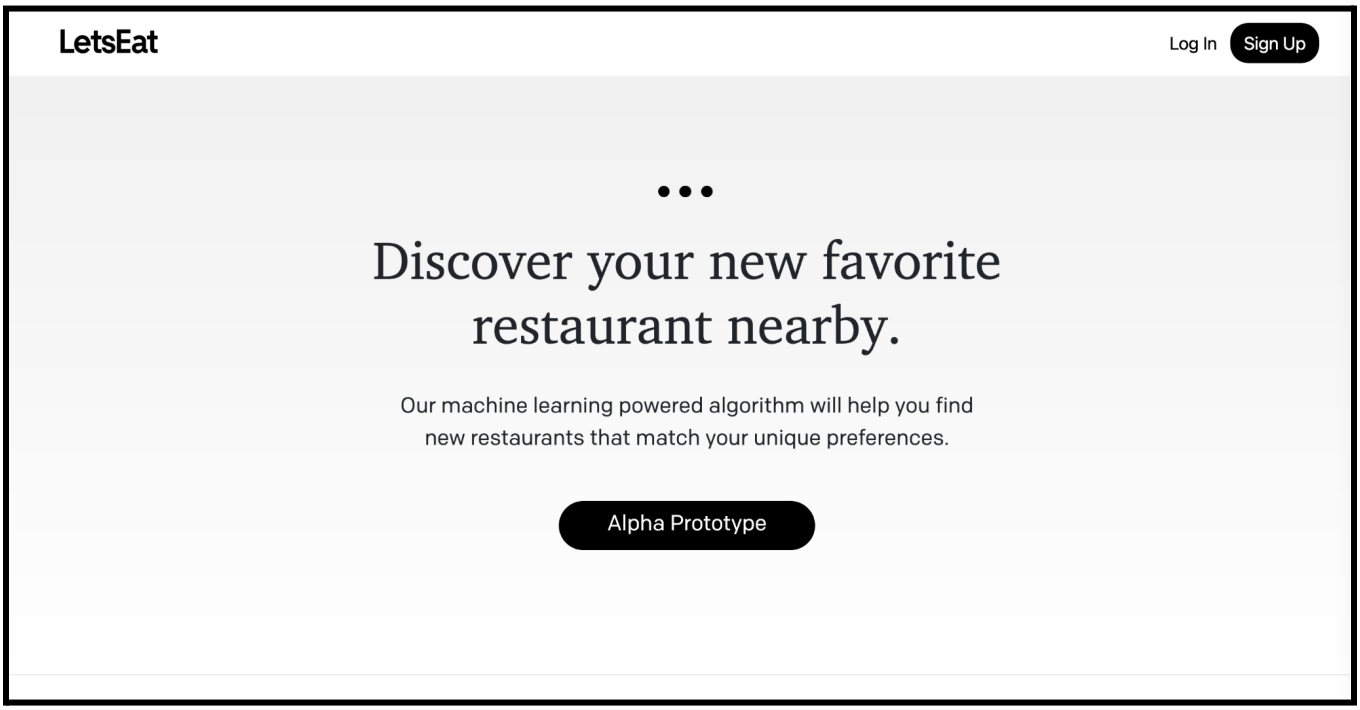
Flow Diagrams

What are the key steps the user takes when interacting with your project? This should be a diagram representation of those steps and what happens upon success or failure of each step.

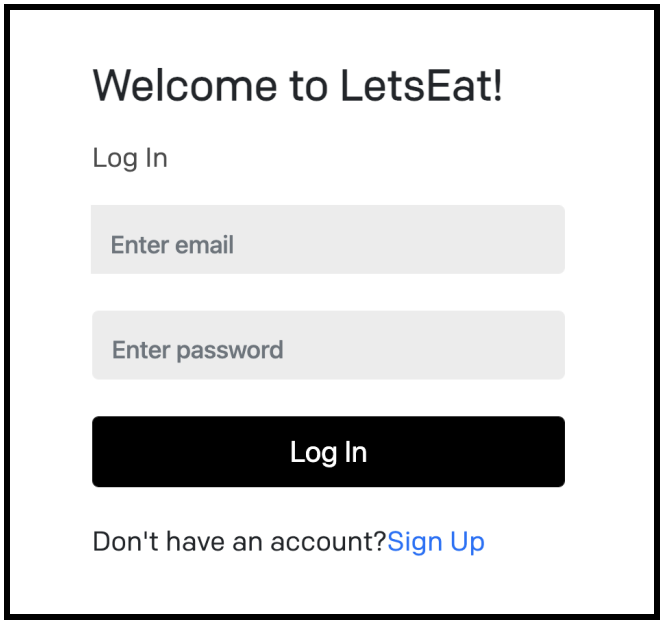


Mockups/Wireframes

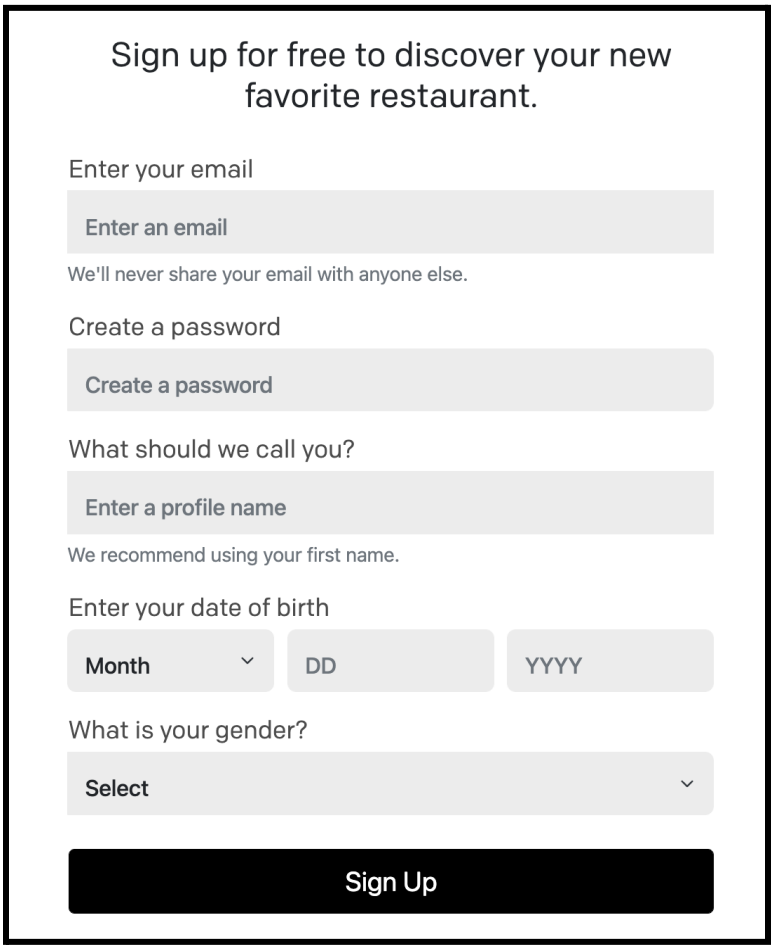
User - Home Page (Mock)



User - Login Page

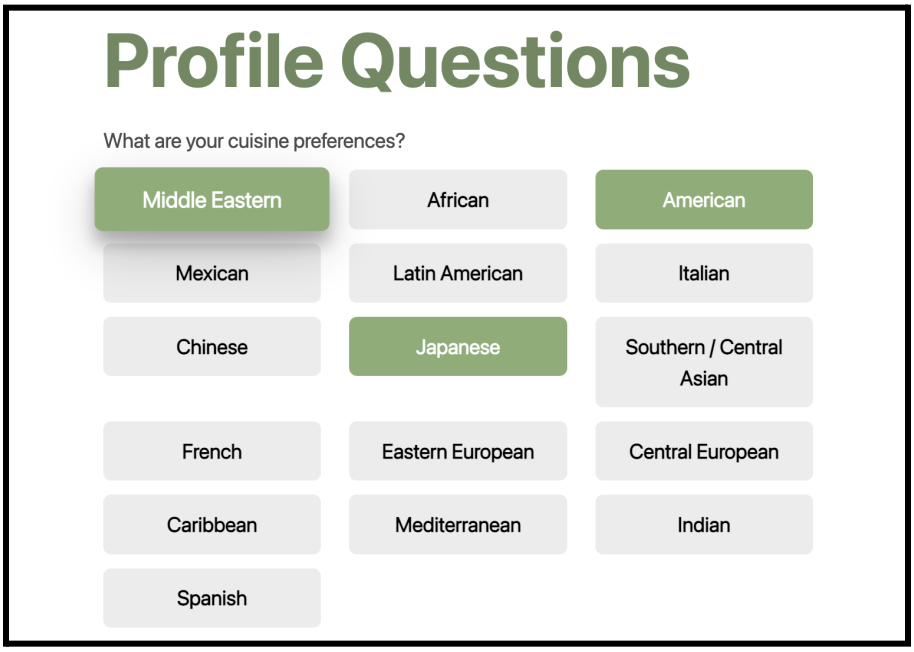


User - Signup Page



User - Search Questionnaire Page

User - Profile Questionnaire Page



User - Restaurant Recommendation Page (Mock)

L'Ardente

Italian

200 Massachusetts Ave NW

Try Again | I'm Going!

Yelp Fusion API Endpoints

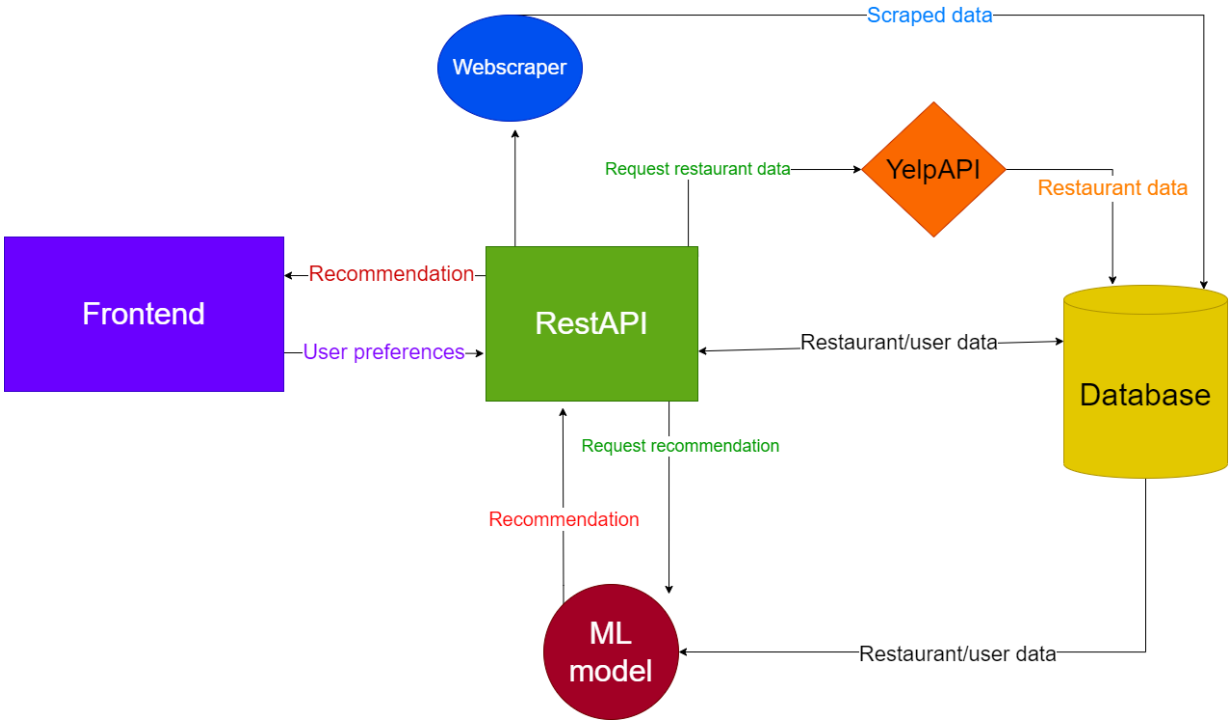
```
headers = {
    'Authorization': 'Bearer %s' % API_KEY,
}
API_URL = "https://api.yelp.com/v3/businesses/search"
```

```
#get list of restaurants based on parameters
def request_businesses_list(zipcode, distance, dollars, open_at, categories, attributes):

    #convert time to UNIX
    #now = open_at
    now = datetime.now()
    unix = time.mktime(now.timetuple())

    #add parameters to API call
    params = {
       'term': 'restaurants', #food vs restaurants
        'location': zipcode,
        'radius': distance,
        'price': dollars,
        'open_at': str(int(unix)),
        'categories': categories,
        'attributes': attributes,
        'limit': 50
    }

    #request API data
    response = requests.request('GET', API_URL, headers=headers, params=params)

    return response.json()
```

```
#use businessId to get json results for that business
def return_business(businessId):
    url = 'https://api.yelp.com/v3/businesses/'+businessId
    response = requests.request('GET', url, headers=headers, params=None)
    return response.json()
```

Technical Specifications

Architecture/System Diagrams



External APIs and Frameworks

- Yelp Fusion API (link): This API gives access to relevant information about restaurants. After the user inputs their dining preferences, data like location, distance, and price range. The API will return the top most relevant 50 or less restaurants and data about them, such as their yelp website, cuisine types, and location. The outputted website will then be scraped to get more information about the restaurant. These restaurants and their data will then go to the machine learning algorithm. This API was chosen because it has the most extensive, detailed list of restaurants and is free to use.
- Selenium Webdriver (link): This framework is used to collect the necessary data to be fed to the machine learning algorithm. The data collection occurs in development, not during user interactions. Using the json data from an API call, Selenium opens a Chrome browser to that restaurant's Yelp webpage. Selenium searches for predefined keywords on that webpage and updates our restaurant database to indicate whether keywords were found. This helps to determine the atmosphere of the website. The Selenium Webdriver also searches the reviews for each restaurant. The search terms include the occasions and allergies a user can enter. The webdriver finds reviews with those keywords, then does sentiment analysis (see Vader) on the review to determine if the user would return to the restaurant. This information is stored in a database and used as training data for the machine learning algorithm.

- Vader Sentiment Analysis (link): The reviews that are scraped (see above) are analyzed and receive a score. This score determines whether the user would return to the restaurant, based on keywords.
- FastAPI (link): This API allows for the infrastructure and setup of the web server for our project. By using this API, prebuilt HTTPS functions can be used to set up API endpoints that the frontend application can ping to send/receive information quickly. FastAPI is just that, fast, and works particularly well with machine learning models so it is a perfect fit for our project which utilizes machine learning to make suggestions.
- XGBoost (link): XGBoost is used to import the decision tree portion of our machine learning algorithm. This API allows for easy setup and use of machine learning models by having them prebuilt and ready to go after import. All that needs to be done once the model is imported is to load it in and fit it with the correct data formatted and encoded. XGBoost was picked for the decision tree model due to its efficiency and speed in training/returning predictions based on user input.
- REACT (link): REACT is the frontend framework used to build the web application for our project. This framework allows for intuitive and dynamic front facing web pages that can change based on user input and the device the user is utilizing to access the web application. REACT was chosen for its ease of use and simplicity; allowing for development of a website that can be simple to use and navigate.

Algorithms

Machine Learning Algorithm:
Train model on combination of made up data and scraped real data in order to give a recommendation that is consistent with individual user preferences and other user recommendations. The algorithm works by first having data compiled by the web scraper and data randomizer to train the model on. This provides a basis for the model to make recommendations as it can refer to assumptions made during training. Once the model is trained, the choices and preferences a user makes are sent into the model. The model compares these preferences to similar users who have similar preferences as well as other restaurants with characteristics that match their preferences. The model will find the restaurants that best fit the characteristics associated with the user and return a list of these restaurants. The top choice is returned to the user as a recommendation, with the rest of the list being additional choices that will be presented if the user rerolls the recommendation, so that computing time is saved.

Additional Algorithm: Web Scraping with Selenium Webdriver
To get the necessary data for the machine learning algorithm, we must scrape the web to get more information about each restaurant and to also generate fake users for training the model. Success in web scraping includes fully populated databases including the relevant information about each restaurant and users for each restaurant. Using Selenium Webdriver, data collection occurs during development. Selenium opens a Chrome browser to the Yelp page of a restaurant, searches for predefined atmospheric adjectives, and updates the database to include this information about the restaurant. Then, Selenium searches the reviews of that

restaurant for keywords about dining occasions and allergies. Sentiment Analysis is done on the review to determine if the restaurant is suitable for that keyword. The database is updated accordingly. These databases are used to train the model and make predictions at runtime.