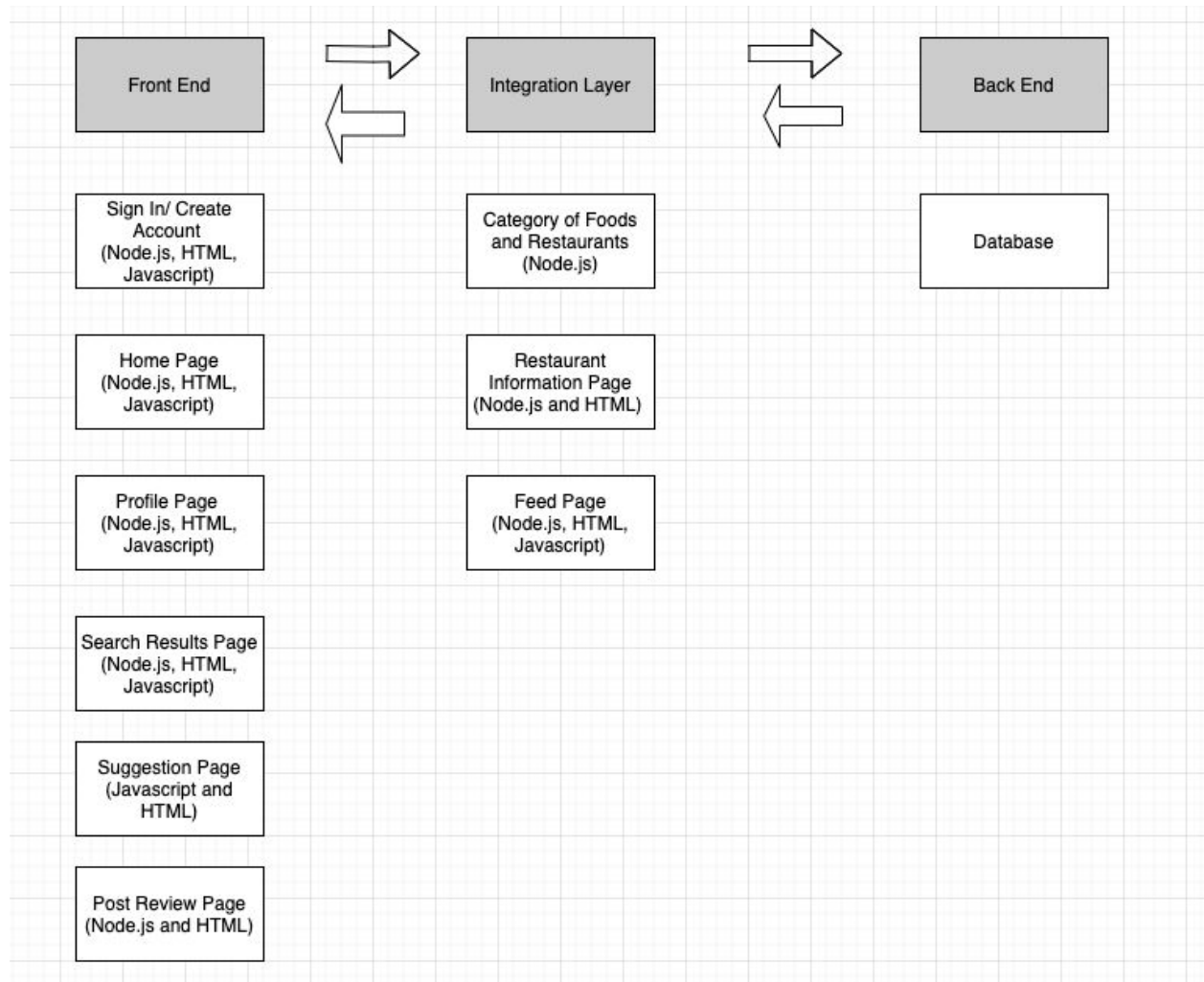


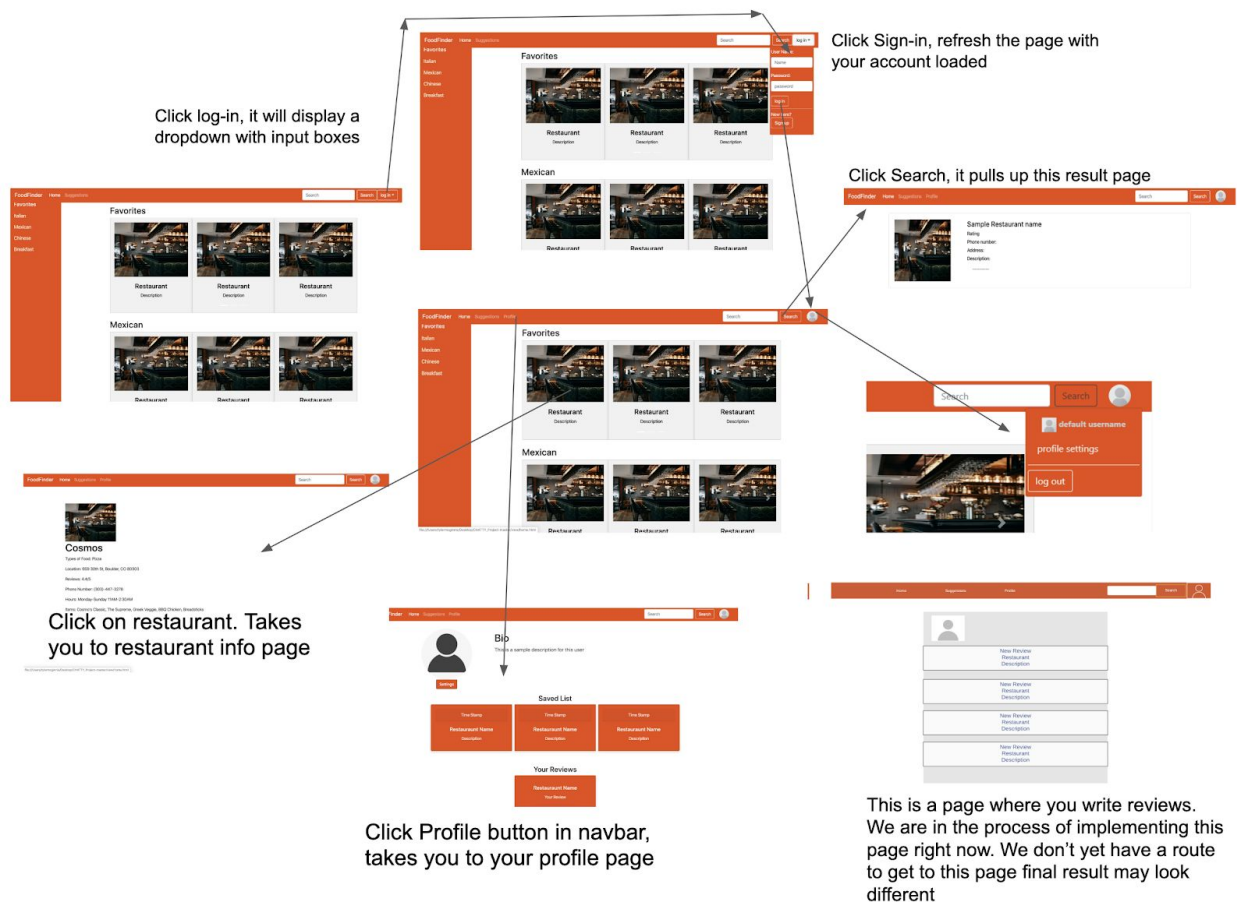
## **Revised List of Features**

- **Sign in/out/up**
  - users can log in or sign up
  - if users are inactive for an extended period of time they will be automatically logged out (already implemented)
    - remember me checkbox for the user to opt out of their log in expiring
  - users may still use the search feature and the home page without logging in
- **Category of the restaurant/food type / User search their food**
  - Once the user signs in, they will be asked what type of food they want. We will have a place where they can select what genre of food (Burgers, Mexican, Italian, Chinese, Fast Food, Dessert, Coffee Shops, etc...). Their selection will help us recommend them a restaurant
- **Recommending Restaurants for people to eat**
  - This is one of the main features of our product. After users select the type of food they want to eat, we will suggest to them a list of 5 restaurants
- **People can post reviews on the restaurants**
  - We want this feature to be similar to Yelp. This will allow users to see what other people thought about a particular restaurant.
- **People can save their favorite restaurants to the save list.**
  - People can save their favorite restaurants so they can have access to promotions and we can use this info to recommend better restaurants
- **Overall Rating System for restaurants.**
  - A way to rank the restaurants so we can suggest the best ones for each user
  - Each restaurant will have a score for the individual user that the database uses to give the user the most relevant restaurant based on the higher score
    - i.e If a user inputs that they want a Mexican restaurant than a restaurant labeled as such appears
- **Photo of menus of the restaurant**
  - Restaurant menus posted on the restaurant's page.

## Architecture Diagram



## Front End design:



Link To Google Slides if this screenshot is hard to see. Click present to view full-screen:

[https://docs.google.com/presentation/d/11Gwk\\_2DWMJIzaM\\_l8GJ-c4rdREW63XV7vi660D9f\\_vs/edit#slide=id.p](https://docs.google.com/presentation/d/11Gwk_2DWMJIzaM_l8GJ-c4rdREW63XV7vi660D9f_vs/edit#slide=id.p)

## Web Service Design:

	description
- post /restaurant/addreview	Pass restaurantid, review content, userid.
- Get /myprofile	gets user info from id stored in cookie and renders the profile page
- Get /home	renders homepage
- Get /feed/{postid}	Pass postid into DB, and get other information back
- Get /feed	Renders feed page
- Post /feed/{postid}/edit	Pass postid and new updated information back to Database
- Get /feed/{postid}/getcomment	Receive comments from the database by postid
- Post /feed/addpost	Pass post content, time, userid into database
- Post /feed/deletepost	Pass postid to database to delete the matching row
- Post /feed/addcomment	Pass comment text, time, userid into database
- Post /feed/deletecomment	Pass commentid to database
- Post /signup	Pass username, password, email to API
- Post /login	Pass username, password to database, receive any matches in and if they match pass the user id to a cookie and set the logged in cookie to true and redirects the page
- Get /logout	sets the csets the values of the log-in cookies to the logged out state and redirects to homepage
- Get /profile/{user}	Get username, email, zip from database using userid
- Post /profile/{user}/update	Pass username, email, zip, etc. to database

- Get /profile/{user}/favorite	Pass userid, and receive favorite restaurantid, and description
- Post /profile/{user}/favorite	Pass restaurants, customerid, description into database.

## Database Design:

- Using PostgreSQL

# Chatty Food Finder

