Mealbot

● ● ●

Group 105-9:
Joseph Petrafeso-Crosby
Malik Tefridj
Huanjian Qiao
Wanglibo Chen

Overview

- Mealbot is a restaurant/meal finder that matches meals to search results and filters based on user preferences like, allergies and dietary choices.
- Our website allows the user to:
  - Login to their account with past information saved
  - Search for restaurants or meals in our database,
  - Change user information (such as username, password, and preferences)
  - Leave a review and rating on a meal

Methodology

- Methodology: Agile
  - We believe this method works best for us since we wanted to make sure that each page does it's own function before we connected all the pieces together.
  - This is also worked well for us because, with an unpredictable school schedule this allowed for much more consistently flexible planning at each stage.
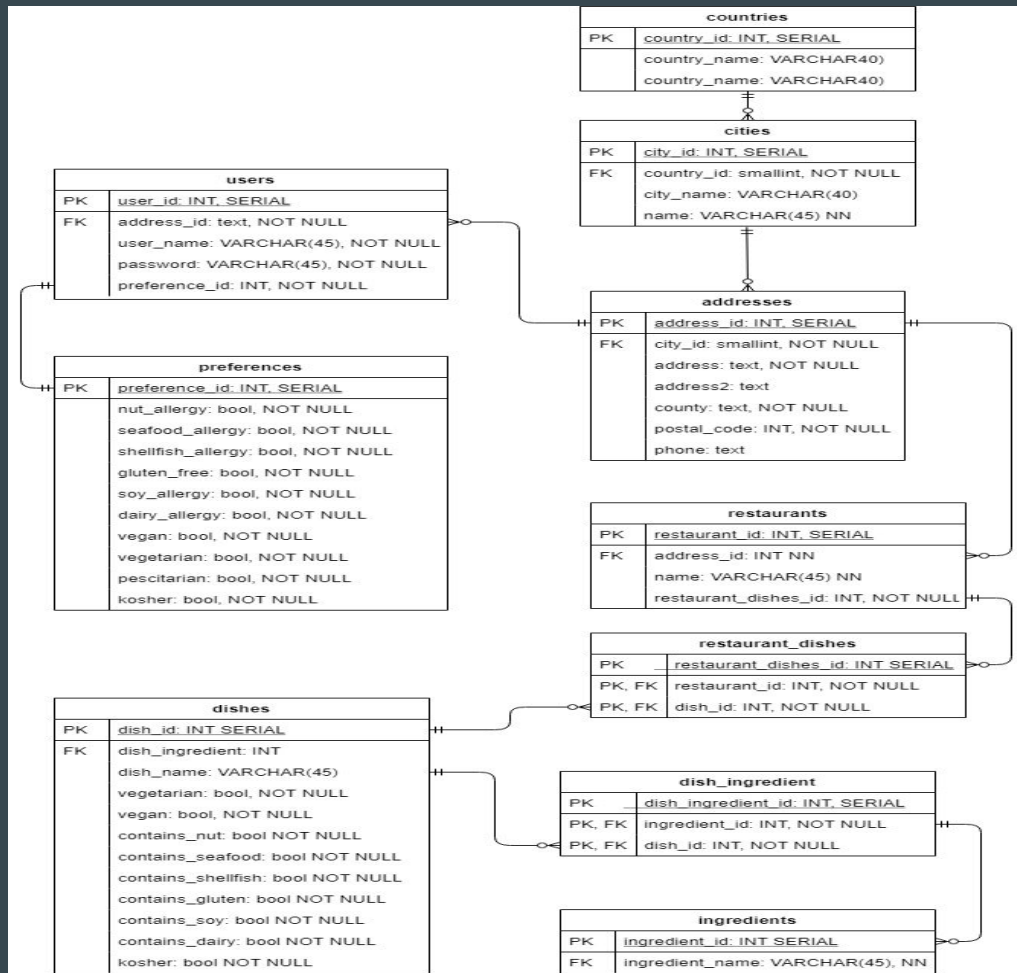 - 4/5

Tools Used

Database

- We are using MySQL as a local database to connect to. All data in the database has been taken from local restaurants from Yelp.
- ⅘
  - Able to edit and view the database schema easily

WORKS WITH MySQL

countries

PK	country_id: INT, SERIAL
	country_name: VARCHAR40)
	country_name: VARCHAR40)

cities

PK	city_id: INT, SERIAL
FK	country_id: smallint, NOT NULL
	city_name: VARCHAR(40)
	name: VARCHAR(45) NN

users

PK	user_id: INT, SERIAL
FK	address_id: text, NOT NULL
	user_name: VARCHAR(45), NOT NULL
	password: VARCHAR(45), NOT NULL
	preference_id: INT, NOT NULL

preferences

PK	preference_id: INT, SERIAL
	nut_allergy: bool, NOT NULL
	seafood_allergy: bool, NOT NULL
	shellfish_allergy: bool, NOT NULL
	gluten_free: bool, NOT NULL
	soy_allergy: bool, NOT NULL
	dairy_allergy: bool, NOT NULL
	vegan: bool, NOT NULL
	vegetarian: bool, NOT NULL
	pescitarian: bool, NOT NULL
	kosher: bool, NOT NULL

addresses

PK	address_id: INT, SERIAL
FK	city_id: smallint, NOT NULL
	address: text, NOT NULL
	address2: text
	county: text, NOT NULL
	postal_code: INT, NOT NULL
	phone: text

restaurants

PK	restaurant_id: INT, SERIAL
FK	address_id: INT NN
	name: VARCHAR(45) NN
	restaurant_dishes_id: INT, NOT NULL

restaurant_dishes

PK	restaurant_dishes_id: INT SERIAL
PK, FK	restaurant_id: INT, NOT NULL
PK, FK	dish_id: INT, NOT NULL

dishes

PK	dish_id: INT SERIAL
FK	dish_ingredient: INT
	dish_name: VARCHAR(45)
	vegetarian: bool, NOT NULL
	vegan: bool, NOT NULL
	contains_nut: bool NOT NULL
	contains_seafood: bool NOT NULL
	contains_shellfish: bool NOT NULL
	contains_gluten: bool NOT NULL
	contains_soy: bool NOT NULL
	contains_dairy: bool NOT NULL
	kosher: bool NOT NULL

dish_ingredient

PK	dish_ingredient_id: INT, SERIAL
PK, FK	ingredient_id: INT, NOT NULL
PK, FK	dish_id: INT, NOT NULL

ingredients

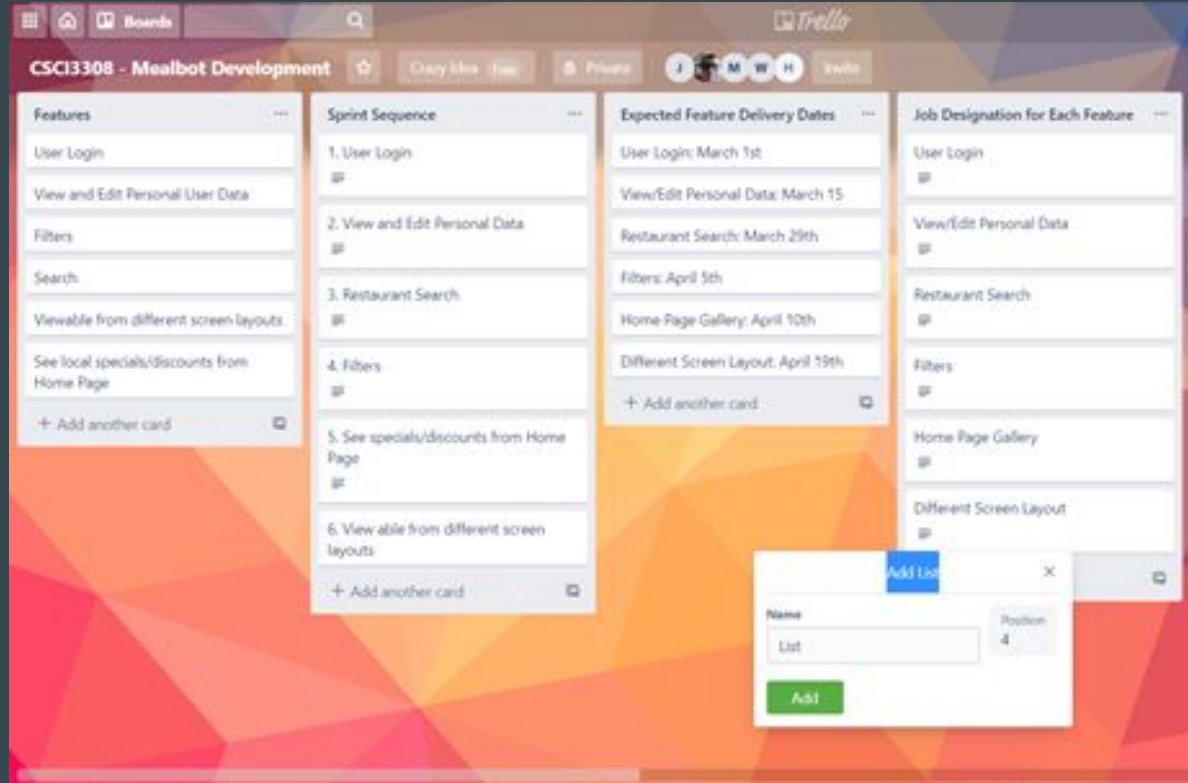|	
PK	ingredient_id: INT SERIAL
FK	ingredient_name: VARCHAR(45), NN

Tools Used

Our Project Tracker

Trello Clipboard

- This allowed us to keep track of various hurdles and milestones.
- However, it was not the most helpful. A quick text message seemed more efficient and certain. 2.5/5

Tools Used

Our VCS Repository

- Github

- This helped us to keep track of different version of file everyone.
- It also made it easier to make sure everyone was was on the same page
- 5/5

IDE

- We use a combination of Pycharm, Atom, and Sublime Text
- 5/5

Testing

- While we didn't use a conventional testing system, we timed how fast our search is, and if the results are different.
- ⅗

Tools Used

Framework

- One of the many challenges we faced was getting Node JS to work with our files so we instead decided to use Python's Tornado server

  3.5/5

Deployment

- We are currently using a local host but are attempting to have our site functioning on heroku
- 2/5

Hardware

- The hardware isn't specific but we have been deploying and testing on our local desktops
- We find this to be less than desirable
- 3/5

Challenges we encountered

- Using new and unfamiliar bootstrap elements, like a multitem carousel.
- Getting our different pages and version to function properly once put together on our server
- Using new and unfamiliar styling methods in CSS