The Gathering Place(s): Database Design

## Overview:

I’ll be using MongoDB for this project to store the data for this project as there are no requirements to link schemas. MongoDB will allow for the storage icon files for menu items and restaurant logos as well inside of documents, and house links to larger outside images when clicking into a restaurant and menu page.

Accessing the data will be performed using an API to communicate between the user-facing part of the website and the persistent storage in the database. Additions to the website data will be queued into separate documents to be reviewed and added by the website maintenance team to review data accuracy and appropriately link existing restaurants and menu items, and verify accuracy of ownership and restaurant menu data.

## Data Structures

This data will be stored in separate schemas as there is rarely a need to load the two connected schemas – restaurant and individual items – together. Those will be accessed together when viewing a page generated by the database after clicking into a search, and otherwise restaurants and menu items can be searched for individually. This approach is recommended by MongoDB to prevent documents from growing and severely impacting performance (MongoDB, n.d.). The simplest way to connect these is to use the unique individual ID, and use $lookups as these will rarely be loaded at the same time and should have a minimal impact on lookup times. Updates will therefore be linked to the original document, and should a restaurant be deleted, calls to retrieve all attached menu items and deal with those will also be included.

Restaurant:

This schema will store information on the restaurant itself – owners, address, directions, and the restaurant’s logo. The entries in this document serve to populate the search for restaurants and to connect to menu entries to provide the name and location of the restaurant they are served at.

// restaurant collection

{

"\_id": "numbers/letters", Generated by Mongo

"name": "restaurant name",

"logo":imagefile,

"largelogo": "link to image file"

"location": "insert address here"

"unavoidableallergens":[ "ingredient one that's everywhere in the kitchen", "ingredient two ..." ],

"owner": "Listed owner if available, back end reference that won't be displayed to public", //optional

"Proof of ownership": "Linked to proof of above, won't be public" //optional

}

Menu Collection  
Contains a list of menu items that have been submitted by owners or general public users, verified and queued for addition by website staff. These entries serve the searches for specific dishes that meet dietary requirements, and provides a linked callback to the restaurant document without being stored inside as an array.

//menu collection

{

"\_id": "numbers/letters", Generated by Mongo

"dish name": "name of menu item here",

"ingredients": [ "ingredient one", "ingredient two", "etc etc" ],

"icon":imagefile,

"largeicon": "link to image file"

"major allergens": [ "shellfish", "milk" ],

"restaurant\_id": "sameIDasinrestaurantSchema"

“verified”: “True or False”

}

Pending restaurants

This document contains the list of forms that have been submitted by restaurant owners, ready to be verified by the back-end staff of the website and approved for addition to the list of restaurants. This is not publicly visible.

//pending restaurant addition collection

{

"\_id": "numbers/letters", Generated by Mongo

"name": "name of User",

"Ownership": uploaded proof of ownership - usually license to operate granted by the city,

"Menu": [ "Manually enter the menu - least preffered method", "dish one", "dish two", "etc etc" ],

"MenuImage": Allows user to upload image instead - preferred method,

"Logo":uploaded imagefile,

"major allergens": [ "shellfish", "milk" ],

"restaurant": "name of restaurant"

"location": "insert address here"

}

Pending Single Items  
This document contains the list of forms that have been submitted by non-owner individuals of dishes they’ve seen, eaten, or have otherwise encountered in the wilds. These will be listed as unverified by the restaurant unless an owner has verified them – but until they are approved they live here.

//pending single item addition collection

{

"\_id": "numbers/letters", Generated by Mongo

"dish name": "name of menu item here",

"name": "name of User",

"photo": Allows user to upload image, //optional

"Logo":uploaded imagefile,

"major allergens": [ "shellfish", "milk" ], //if known, user uploaded

"restaurant": "name of restaurant"

}

Users – staff

The database of users that can log in as staff, approve additions to the restaurant or menu lists, or delete listings if necessary. (A note – I was working through the workflow of how this would be updated and realized we will need a backend with some automated functionality or this will just be tedious. Thus, users and a login.)

//Users

{

"\_id": "numbers/letters", Generated by Mongo

"name": "name of User",

"username": "User of name"

"password": "passcode for logging in"

}

## Stretch

In the eventuality of adding a forum to the website, we will need ways to store the data that goes in the forum, and the users of the forum, and similar. That will be these:

//Forum Users

{

"\_id": "numbers/letters", Generated by Mongo

"Email": "email of User",

"username": "login and screen name"

"password": "passcode for logging in"

}

//Topics

{

"\_id": "numbers/letters", Generated by Mongo

//Posts

{

"\_id": "numbers/letters", Generated by Mongo

"Topic": "Topic under which post falls, tied to topic ID"

"User": "They who created the post, tied to ID of user"

"Post Title": "String, title of post"

"Post content": "Very long string, content of the post"

"Replies": ["Array of replies", "Capped at about 256 replies", "Contains strings of replies"]

}