# Food Trucks Nearby

## Database Overview

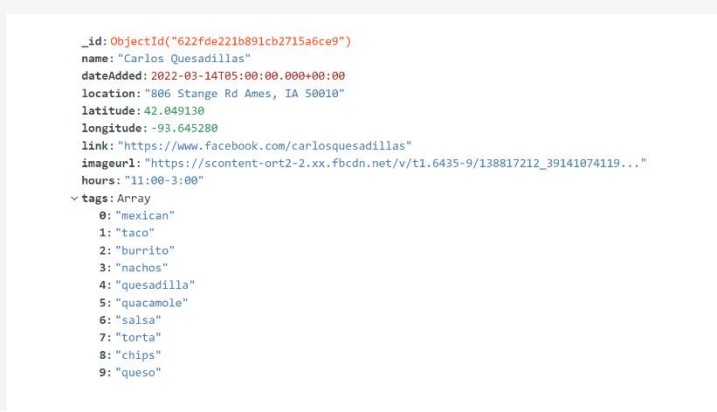
For my project “Food Trucks Nearby” I plan on using MongoDB to persist data. I believe that a document database design such as MongoDB will fit my application well. Currently I have one relationship between data, and that is in the user favorites array and a food trucks id. I do not believe this one relationship justifies using a relational database. Beyond this relationship each document will contain all the relevant data necessary for each specific object. The free cloud hosting and additional features such as monitoring further make MongoDB ideal. These features could be useful as the application expands and grows.

I am currently considering using Mongoose to create a schema for the food truck cards. The cards do not require all the information features in the food truck collection objects and therefore need to be limited for each card. Furthermore, Mongoose may be useful for future feature releases.

## Data Structures

### Food Trucks

```
{
   “_id”: string (autogenerated by MongoDB),
   “name”: string,
   “dateadded”: date,
   “location”: string,
   “latitude”: string,
   “longitude”: decimal128,
   “linkurl”: string,
   “imgurl”: string,
   “hours”: string
   “tags”: array [
      {string},
      {string}…
   ]
}
```




### Users

{
“_id”: string (autogenerated by MongoDB),
“fname”: string,
“lname”: string,
“email”: string,
“password”: string,
“favorites”: array [key identifier
{“foodtruckId: string},
{“foodtruckId”: string…
]
}


# Data Context & Implementation

## Food Truck Collection

The core of the application is built around the data in this collection, making it the heart of the application. Without it the application would be empty and useless. The food truck collection data is used to populate the food truck cards that are found in the favorites list and the food trucks page. The data will be populated using a server API between the application UI and the database, so users will not have direct access to the data or

database. In future builds this same data can be added to or used in other ways, including related truck cards and full pages for each food truck.

The application search function will filter through the data stored in the tags array to return refined results to a user. The data field of “dateadded” will allow the application to sort food trucks by most recently added or oldest.

## User Collection

The data stored in the user collection is used regarding specific users. The objects are created when a user creates an account, and the stored data is used to validate sign in information. The user object will be used to associate a specific user with the trucks they have added to their favorites. My current plan is to have the favorites button PUT and DELETE accordingly to the favorites array.

This collection has the only relationship wherein the ids stored in the favorites array reference the ids of food trucks in the food truck collection. This allows users to save favorites and for those favorites to indicate as such, both in the favorites section and on the food trucks page.

## Questions/Thoughts for Feedback

For my food truck collection, I would like to eventually be able to store different hours of operation. While many food trucks have usual hours and locations, sometimes they will wander or travel to events. I am uncertain as to how I would store and retrieve the data. Some thoughts I have had would be a key (ie 0, 1) with a different string of hours for each. If I could get the system to associate a day with a specific key (ie Sundays are the hours at key 0), This is not necessary for the MVP but I am curious nonetheless.

A thought I have had would be adding the name of the food truck within the tags data array, both in separate words and a single string. My thinking is that this would make it easier for users to search for the name of a food truck in addition to other keywords.