Food Voter App Database Design

Andrew DeLeon

SWDV 691

October 27, 2021

Introduction

When looking into the type of database to use with the Food Voter App, it seemed that using more of a Document Database made more sense than a Relational Database. Reason being is because the app is meant to be more simple and easier to gather the information. The application, starting out, will only be taking in the user’s information, the voting information, and the types of food that the users are voting on to decide on where to eat. A Document Database would be able to hold this information in a better way because there are not many different relationships occurring since it is a smaller app.

Researching this topic further, I have decided to use MongoDB for the main database. MongoDB would be the best option because it is a Document Database that is scalable to the application’s needs. MongoDB also is easier to change over time. With the Food Voter App having potential to grow later, I want to make sure I have a good database that is going to grow with the app. Below is an example of how the Document Database would be set up for the Food Voter App.

Example

The User Information is the information that will be gathered with a login.

User Info:

```
{"id": "1",
 "first name":"John",
 "last name":"Smith",
 "email":"jsmith@email"}
```

The food information is taking the name of the food. This is information will be taken into consideration when the voting is being completed.

Food info:

```
{
    "food id": "1",
    "food type": "American"
}
```

The vote is where most of the information is being pulled from. The vote is taking the user information and the food information and then keeping track of the user vote status.

Vote:

```
{
   "vote id": "1",
   "user id": "1",
   "food id": "1",
   "user first name": "John",
   "user last name": "Smith",
   "food type": "American",
   "vote status": "Done"
}
```