Ryan Hilton
SWDV 691 – 1W 18
November 2018

Database for Flexible Cookbook App

This database design has been updated to include a stretch feature, which is images of specific dishes that are stored with the recipe. The structure will be that of a document design, using MongoDB. This strategy was chosen because there is expected to be several one-to-many relationships, finite data, and considerably more read requests than write requests.

The document collections for the MVP have been identified as the following: user documents, conversion factor documents, recipe documents. For the stretch feature, the recipe documents will have an additional field for images, with an array of key/value pairs. These key/value pairs will have the name of the image and a reference to where the image is stored.

Each document will be in JSON format, and is expected to have limited bounds. The document specifications and examples of each document collection type are shown below:

User Documents :

- Will contain the following key/value pairs:
  - "userid": (integer)
  - "name" : (string)
  - "recipes": (array of strings)
  - "recipe_ids": (array of integers)
- The array of recipe names will correspond to the array of recipe ids, so that multiple users can have the same name for a recipe, but it corresponds only to their own recipe id.
- Will have a strategy for handling relationships – a field with an array containing the identifiers for each recipe saved by the user.
- Can be scaled to support secure login later, but for MVP this will suffice.

Recipe Documents:

- Will contain the following key/value pairs:
  - "id": (integer)
  - "instructions": (string)
  - "ingredients": (nested array of key/value pairs)
    - [{ (ingredient_name(string)): quantity(float) }, { . . .}, {. . .}]
- Can be any size needed, as required by the number of ingredients listed.
- If a recipe gets edited, it will delete and replace this entire JSON document.

- For the stretch feature, these documents will also have the following key/value pair:
    - “images”: (nested array of key/value pairs)
        - [{image name(string): reference to location(string) }, { . . .}]

Conversion Factor Documents:

- Will contain the following key/value pairs:
    - unit_of_measure(string): conversion_factor(float)
- The “conversion_factor” field is the number by which this measurement must be multiplied in order to convert the “quantity” (from recipe document) into the actual amount used in the recipe.

These documents will be available to the service layer in the form of JSON replies, containing the data requested. In the case of a recipe document, the entire document will be retrieved and used to populate the fields required by the user interface.

Examples:

User documents:

```
{
   "userid": “001”,
   "name": "Tobias",
   "recipes": [ “Cake”, “Banana Bread”, “Pizza” ],
   "recipe_ids": [ 003, 006,  007 ]
},
{
   "userid": “002”,
   "name": "Martha",
   "recipes": [ “Pie”, “White Bread”, “Pizza” ],
   "recipe_ids": [ 001, 005,  008 ]
}
```

Recipe documents:

```
{
   "id": “001”,
   "instructions": "Mix flour and water together in a bowl . . .",
   "ingredients": [
                  {“Flour”: 3.5},
                  { “yeast”: 0.002},
                  { “Water”: 3.15},
                  { “Eggs”: 3}
                   ],
```

```
“images”: [                                          STRETCH FEATURE PORTION
        {“First try”: “/images/image1.jpg”},         STRETCH FEATURE PORTION
        {“Perfect”: “/images/image2.jpg},            STRETCH FEATURE PORTION
        {“Jason’s one”: “/images/image3.jpg}         STRETCH FEATURE PORTION
        ]                                            STRETCH FEATURE PORTION
}
```

Conversion factors document:

```
{
   "Cup": “1.00”,
   "Pint": "2.00”,
   "Quart": “4.00”,
   "Gallon": "16.00”,
   "Ounce": “0.125”,
}
```

Preliminary research indicates that the image files for the stretch feature are expected to be less than 16 MB in size. Therefore the intention is to store them directly in the database using the BinData data type. If additional research comes to indicate that the image files may be larger than 16MB, another solution for storage will be implemented, such as GridFS.