Design: Database

I've decided to use the Document Database Design for the Salon Locator app because it forms its own structure. In addition, many different types of documents can be mixed together in a document store. This makes document database design much more flexible since the structure does not have to be consistent. Adding new information is much easier since new salons and spas may be added frequently, and a salon's information could change often.

Source: <https://www.ionos.com/digitalguide/hosting/technical-matters/document-database/>

A Document Database I'll use for this app is MongoDB. There will one collection that will be automatically created whenever one of the following takes place:

* a user registers for an account as either a customer or a salon owner
* a user who signs up as a salon owner enrolls a salon into the app
* a salon that is has been enrolled is a hair salon, a nail salon, or a spa
* a user, whether he or she is a customer or a salon owner, 'Favorites' a salon, which forms a 'Favorites' list
* a salon that has been added onto a user's 'Favorite' list earns a 'Heart'

Within the collection there will be a document with several subdocuments, and MongoDB forms the collection once data is inserted. I have written an example of a collection with documents and subdocuments below:

{

first\_name: “Lucy”,

last\_name: “Martinez”,

username: “lucy\_M”,

email: “lucy\_M123@yahoo.com:,

address: {

no: “123”,

street: “Alphabet Lane”,

city: “Atlanta”,

state\_or\_prefecture: “Georgia”,

country: “United States”,

location: 0000000,

}

account\_type: [“customer”, “salon owner”],

salon: [

logo: “<image\_placed\_here>”,

name: “Lucy’s Dominican Salon”,

salon\_address: {

salon\_no: “2359”,

salon\_street: “Windy Hill Road SE”,

salon\_city: “Marietta”,

salon\_state\_or\_prefecture: “Georgia”,

salon\_country: “United States”,

salon\_location: 0000000,

}

days\_of\_operation: [“Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”,

“Saturday”, “Sunday”],

hours\_of\_operation: {“<document>”},

services\_provided: {“<document>”},

hearts: 2,

salon\_type: [“hair salon”, “nail salon”, “spa”],

],

favorites : {

salon\_name: “Lucy’s Dominican Salon”,

}

}

The fields that are colored pink are string data types

The fields that are colored green are Geo-coordinates

The field(s) that are highlighted yellow are integer data types

The fields that are colored blue contain subdocuments containing stings or are arrays that hold strings

There is a field that may be a null data type since it requires the user to upload an image