

Database Type Justification

For Image Portal, I have carefully evaluated the type of data that is going to be stored, how the data will be used, and the potential relationships between different data types. When considering the data store type requirement, most of the information is strictly going to either be basic key value pairs, such as account info, or meta data for photos, such as image name. The data will be used by the Image Portal for basic CRUD operations, with an emphasis on reading, as this will be used by the API to display data on the site. There will be some data fields also associated with the user account, but those are only intended to be used in the login process.

Given the above considerations, a document database such as MongoDB would be ideal for this project. Each user can have a discrete Image Portal User document object in the database and during the MVP there is no intent to create relationships between documents. In the future if other features such as collaboration are implemented, additional document relationships can be evaluated.

Speaking specifically in terms of how the images are going to be stored, the best option is likely a cloud storage such as Amazon S3. An alternate option would be to manually store these images in the MongoDB database, but having the flexibility without the complexity of MongoDB image storage is enticing.

Data Structure

Image Portal User Document

Any fields highlighted in **blue** are for STRETCH feature(s) only and are not included in the MVP

Keyname	Data Type	Usage
_id	ObjectId	Generated, Primary Document Key
username	String	username for logging in
password	String	password for logging in
firstName	String	First name for user logging in
lastName	String	Last name for user logging in
Location	String	Location for user logging in
Bio	String	Bio for user logging in
twitter	String	Twitter link for user logging in
linkedIn	String	Linked In link for user logging in
personalWebsite	String	Url for personal site of user logging in
userIcon	String	Icon url for image; hosted in S3
shareAbleData	Array of String, Boolean	Selection of what is being shared by key name
albums	Array of albums	contains n album objects to support multiple albums
albumId	Integer	unique id for album, generated when created
albumName	String	album name, required when user creates album
albumCreated	Date	album creation date, displayed in UI
images	Array of images	contains n image object
imageId	Integer	unique id for image, generated when uploaded

imageName	String	image name, required when user adds image
imageTag	String	image tag, used when organizing images
imageUpload	Date	image upload date, displayed in UI
imageLocation	String	image URL, used when performing any CRUD operations including email link

JSON Format

```
{
  "_id": "",
  "username": "test123",
  "password": "",
  "firstName": "M.J.",
  "lastName": "Carrington",
  "location": "St. Louis, MO",
  "bio": "I'm a SDET from st. louis!",
  "userIcon": "https://s3.amazonaws.com/f73123123.png",
  "twitter": "http://www.twitter.com",
  "linkedIn": "http://www.linkedin.com",
  "personalWebsite": "http://www.google.com",
  "shareAbleData": [
    {
      "firstName": false,
      "lastName": false,
      "location": false,
      "bio": false,
      "twitter": false,
      "linkedIn": false,
      "personalWebsite": false
    }
  ],
  "albums": [
    {
      "albumId": 12345,
      "albumName": "Skating Event",
      "albumCreated": "2021-03-23T01:26:20.112Z",
      "images": [
        {
          "imageId": 1,
          "imageName": "name here",
          "imageTag": "tag image here",
          "imageUpload": "2021-03-23T01:26:20.112Z",
          "imageLocation": "https://s3.amazonaws.com/TEST/f82dd4ac221b.png"
        },
        {
          "imageId": 2,
          "imageName": "name here 2",
          "imageTag": "tag image here 2",
          "imageUpload": "2021-03-23T01:26:20.112Z",
          "imageLocation": "https://s3.amazonaws.com/TEST/f82dd4ac221b.png"
        }
      ]
    }
  ]
}
```