*Nikhil Ranjan*

*\*If missing anything please add and put a small explainer in the group chat about it, thanks. We should be able to follow these steps to finalize a database good enough for our needs. \**

**Define data types:** Identify the types of data that your web app will handle, such as images, videos, audio, and other unstructured data. Determine how you will store this data in your NoSQL database and what kind of data structure is appropriate.

* Images
* Videos
* Audio files
* Structured – relational data
* Unstructured – storage of cracked captcha? Need clarity

**Database type:** NoSQL databases come in several types, such as document-oriented, key-value, column-family, and graph databases. Choose the type of database that is best suited for your data and application requirements.

* Mongo DB - is Document oriented but can also store images,audio and videos
  + Dynamic DB – non relational
  + Mongo DB uses collections – equivalent to a table in a relational DB. A collection is made of documents, which is equivalent to a record in a relational DB table.
  + Each document will store a user and consist of other information which will make them unique such as an ID, first and last names and email.
    - Will also contain data associated with each user –
    - Each data will have a piece of ID associated and have the User ID
  + Data is stored in JSON format so it is easier to work with using python framework

**Define data model:** defining the entities (e.g., users, products, images) and their relationships with each other, and defining the attributes and properties of each entity.

Data will be Embedded ( single document/structure) …

* ML Model
  + Model ID
  + Model Type
  + Captcha Type/s
  + History collection?
* Users
  + ID
  + First Name
  + Last Name
  + Email
  + Username
  + Password?
  + Date
  + Data
    - Images
    - Videos
    - Audio

**Determine access patterns:** Determine how your application will access the data in your NoSQL database. Consider the types of queries your machine learning models will execute and design your database schema to optimize those queries.

*(Basically, use cases)*

* Web based client –
* What are the queries of the DB?
  + User wants to see the progress of their captcha breaking on their account.
  + User wants to upload a set of captchas in their own account.
  + User wants to see statistics of individual captcha breaking.
  + User wants to see statistics of a class of captchas being broken.

**Optimize for scalability:** Consider sharding, replication, and other techniques for scaling your database as your application grows.

**Configure security:** Configure authentication and authorization for user accounts. Use MongoDB's built-in role-based access control (RBAC) to restrict access to sensitive data.

(Choose indexing strategies? Maybe not needed)