

1.

Our cache uses MongoDB to cache the most recent 100 search results of anyone using our app. The searches and their results are saved in a key:value pairing in an array of arrays. When a user calls our search function, the function will first scan the array for the search term and return the results if the key was found in the array. After 100 searches, we intend for the first value of the array to be deleted (as we are appending values to the end of the array) before adding the next value.

What we have not implemented but are looking at is a time-based expiration for some of the entries in the cache, because incorrect searches could be made if a search term, say "creator:Grateful Dead" is called repeatedly. Our app would never be able to update its search for the Grateful Dead entry. This would be relevant if new results appear on the archive.org search.

Our code is fully implemented on our team's GitHub repo.

2.

Cache Representation

```
db.createCollection("cache", {  
  validator: {  
    $jsonSchema: {  
      bsonType: "object",  
      required: [ "query" ],  
      properties: {  
        query: {  
          bsonType: "string",  
          description: "a string, which is the search term used by the user, required"  
        },  
      },  
    },  
  })  
})
```

```

db.createCollection("userdb", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: [ "id, favorites`~~, " ],
      properties: {
        id: {
          bsonType: "string",
          Description: "string id of the user, required
        },
        Favorites: {
          bsonType: "array",
          Description: "array of favorite bands saved by user, not required"
        },
        History: {
          bsonType: "array"
          Description: "array of recent searches for easy access, not
          Required"
        }
      }
    }
  }
})

```

Deadhead



Our App
and
Database

Google



Archive.org

