**Design Document of Node.js Blog**

Yuefeng Ju

20086447

***1.Web App Functionality***

1. **Server Function** This restapi-style blog has basic features.
2. Management of Administrator: Like many popular blogs today, administrator management is the most basic. As a developer, I added the admin backend add, login featureIt is worth mentioning that my password transmission method uses md5, which will enhance the security of users. The user logs in with the username and password added in the background, and then can implement various functions. Users can log out after use, after logging out, they can view other people's blogs and news but cannot comment on them.
3. Management of Category: Users can add categories to their blogs before writing a single article. Users can add categories, and if they do not meet the character limit, they will be prompted with an error. Users can view the created categories and can specify to update their names. Finally, the user can delete the specified category. Of course, they can also see how many blogs exist in each category.
4. Management of Blog: For the main part of the blog, the user can post the blog. They need to add a title, a body, and select a category from the created category and pick a tag (there are currently 4 tags: paper, blog, news, twitter, you can add tags from the background), among them, I limit the number of words in the title to 50 words. On the homepage, users can see all the blogs that are visible for publication, arranged by the time of publication and can see the title, short text and author information
5. Review of Blog: Adding comments to other people's blogs is very interesting. When users log in, they can add comments to others or their own blogs. Of course blog owners can delete comments. Comments will show the time and author of the comment and the content of the comment.
6. News Show: Through third-party APIs, users can view the latest news about Apple Inc. Similar to blogs, on the news page, they can see the title, short content and author. When they click to enter, they will automatically jump to the designated news page for detailed information.
7. Get weather: Through a third-party api, the web app can automatically obtain the current weather in Waterford (displayed in Fahrenheit and weather conditions)
8. **Client Function**

client

\* news page: guest can browser news about 'apple' in the nearest week, the titles can be clicked and linked to the external website.

\* home page:

+ published blogs are displayed in this page's list.

+ guest can click each title to go blog's details page to read more.

\* details page: if logged in, user(administrator) can add review, if not logged in, guest can only browser the details.

\* category page

+ the added categories are listed as label.

+ logged-in user can add new category and remove category, but guests cann't.

+ the sorted blogs by categories and their counts are showed in a table.

+ guest can click the category's link to the blog's list page, namely home page, this page only show the blogs of the clicked category.

\* blog page

+ there is a table to show the blogs' title, category, tags...

+ logged-in user can delete the blogs.

+ user can click each title's link to see the details.

+ click the post blog button, this page would go to the form page.

\* form page: user can post a new blog. there are title, content, category, tags fields.

\* login page

+ after clicking the login button, user can see this page.

+ name and password are required to login.

+ after logged in, the browsered page before would be showed.

\* navigation bar

+ there are HOME/NEWS/CATEGORY/BLOG four tabs to link each page.

+ there is a text area to show the current weather of Waterford.

+ there is a button of LOGIN to login or the logged-in user's name and LOGOUT button to loggout.

***2. Specific framework***

#### back-end:

\* Node.js

\* MongoDB Atlas

\* Express

#### front-end:

\* vue / vuetify / vue-router / vuex

\* cypress

\* sass

\* axios(to request)

***3. 3rd Party APIs***

\* Current Weather of Watorford

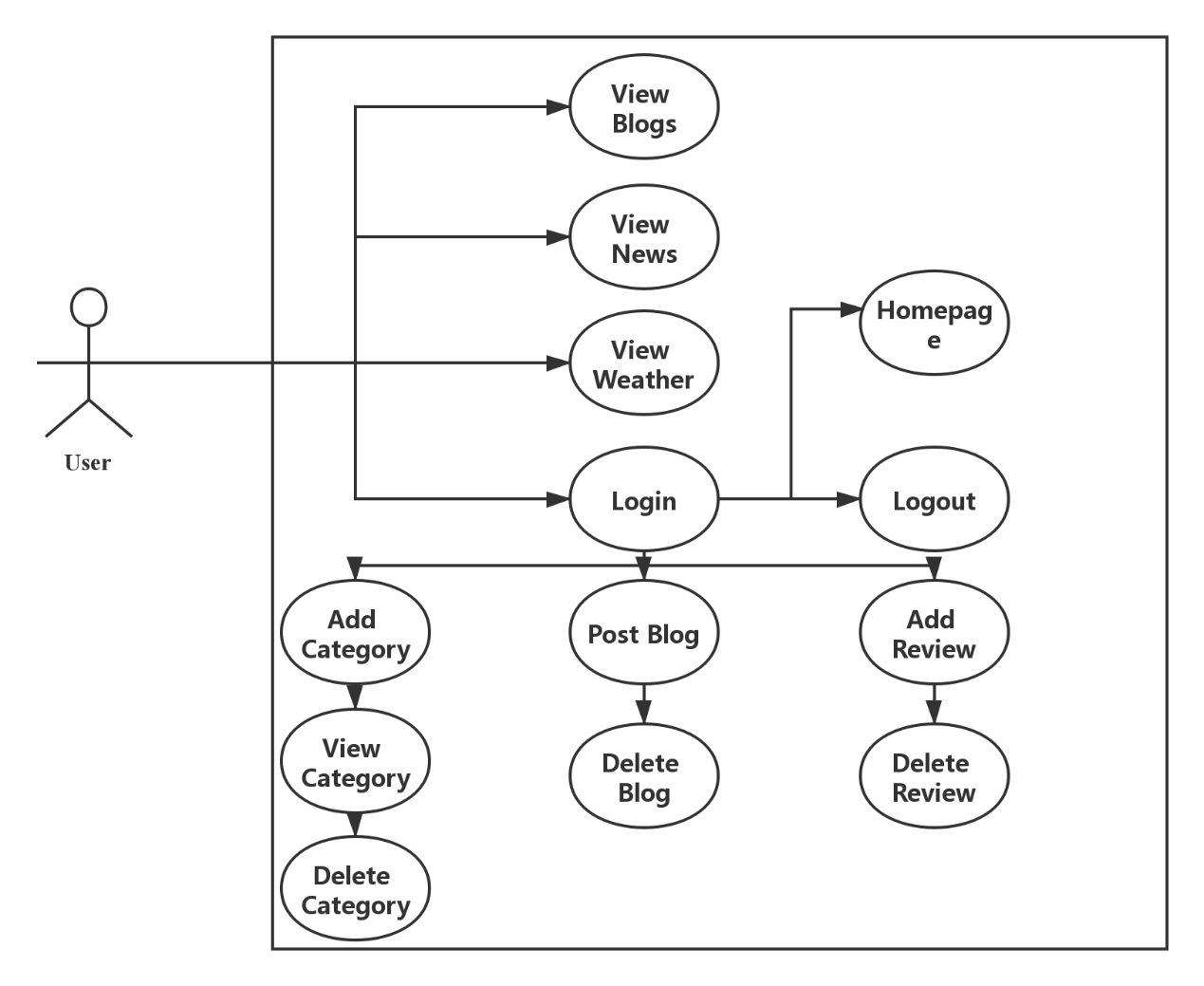
<https://api.openweathermap.org/data/2.5/weather>

\* News of Apple

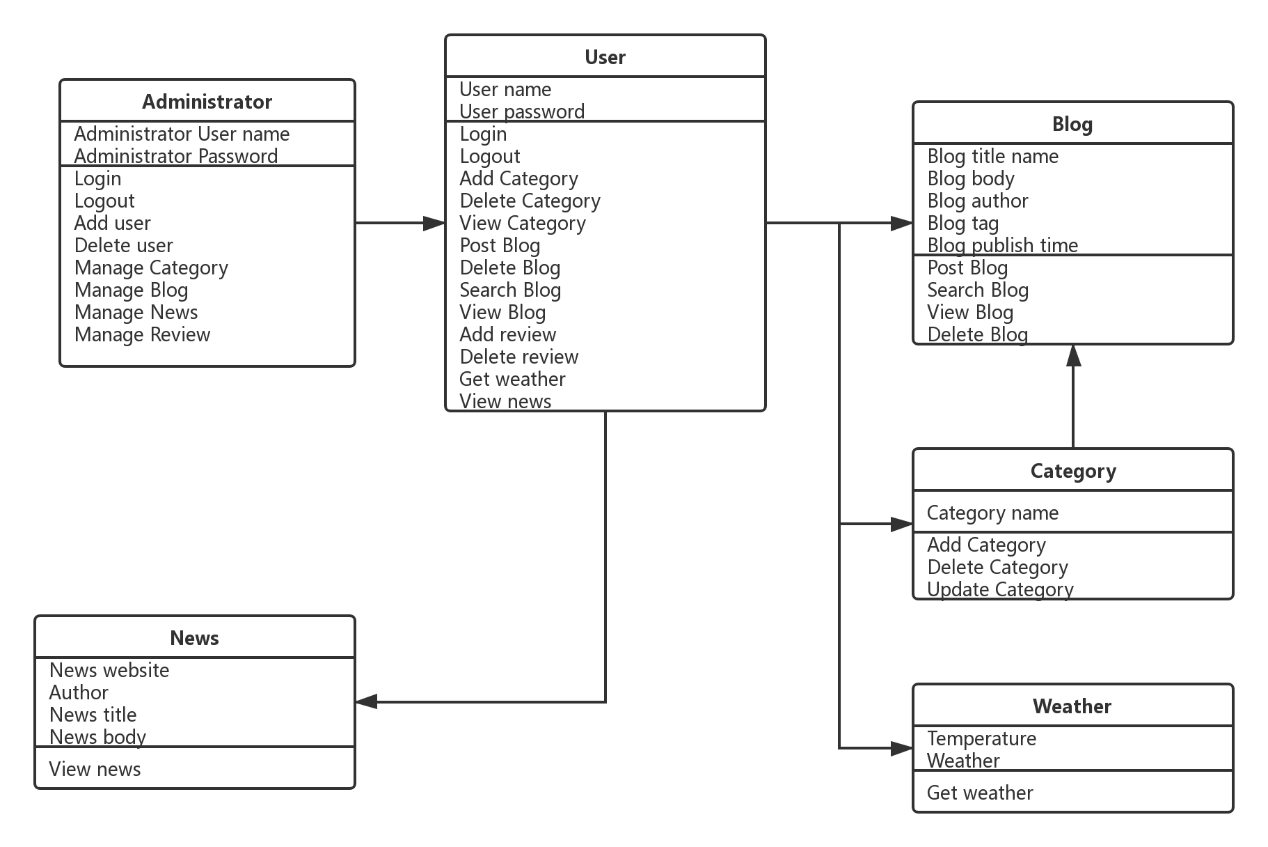
<https://newsapi.org/v2/everything>

***4. Appropriate UML Diagrams & Use cases***

1. **Use Case Diagram**

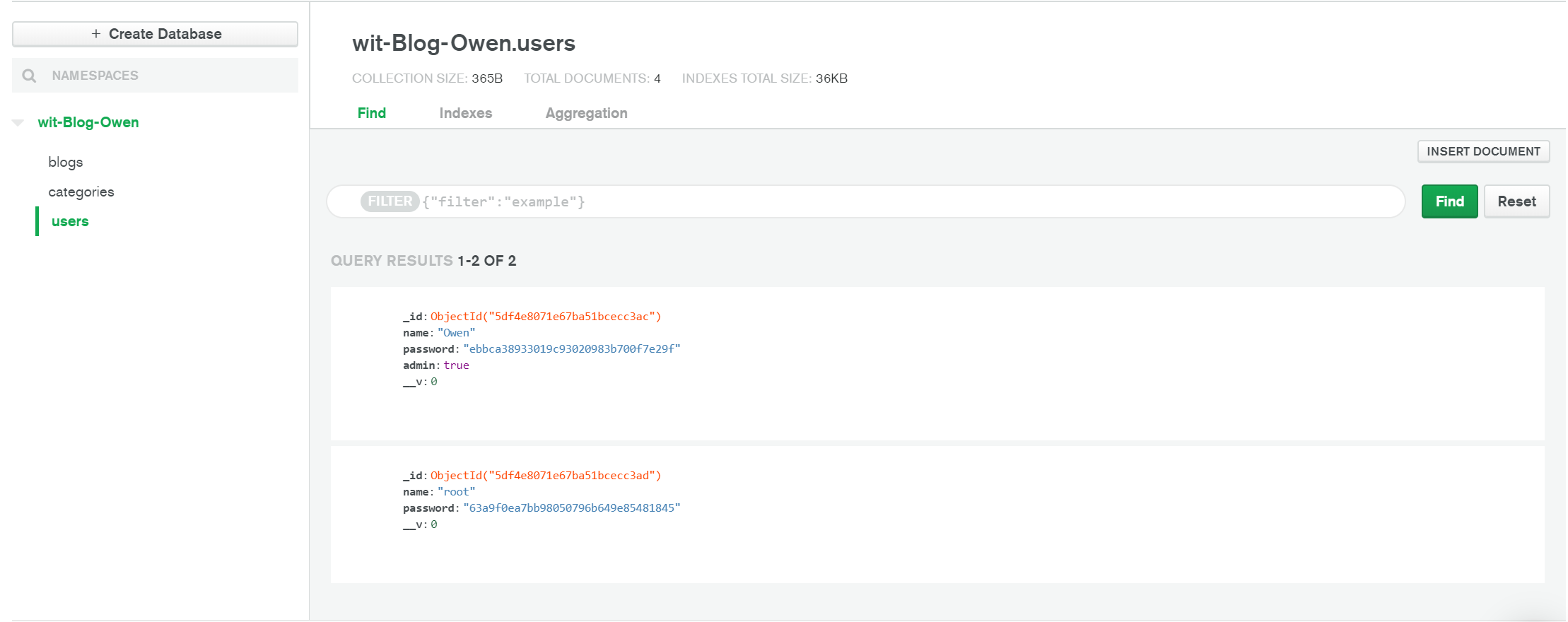


1. **Class Diagram**

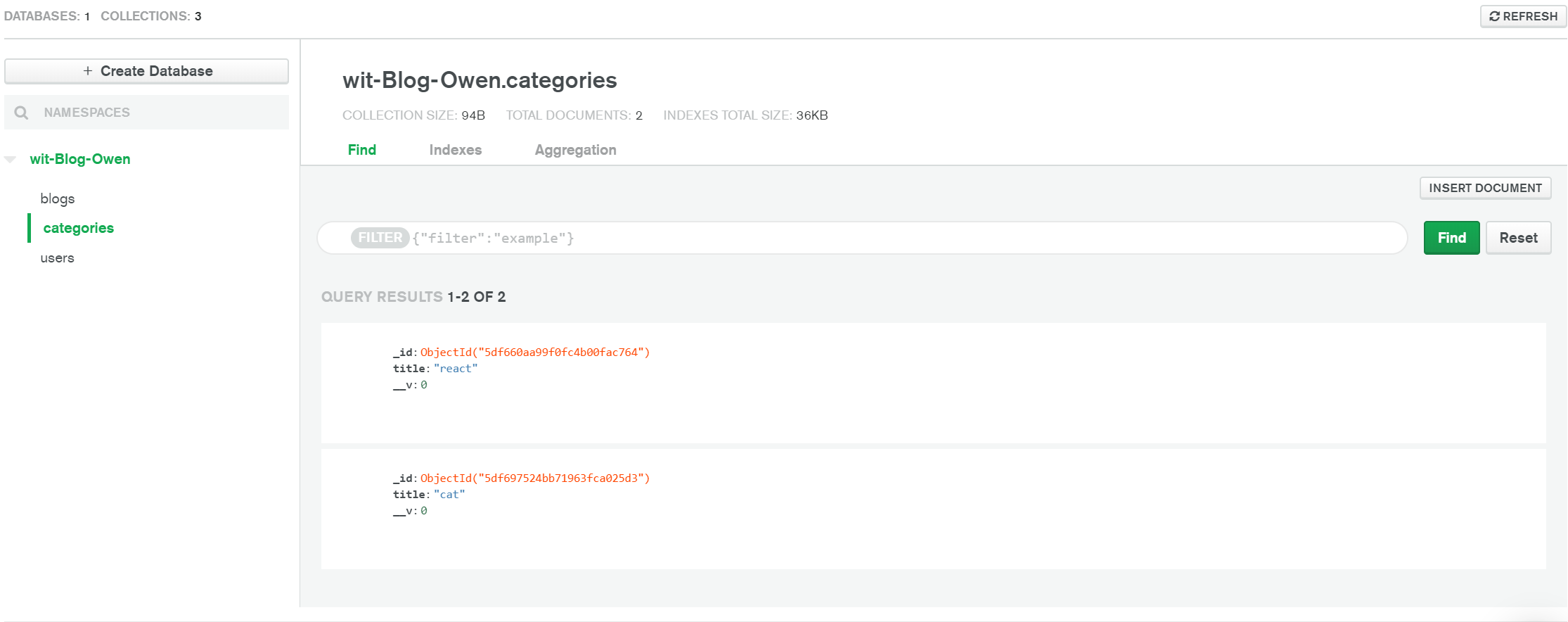


***5. Database Schemas***

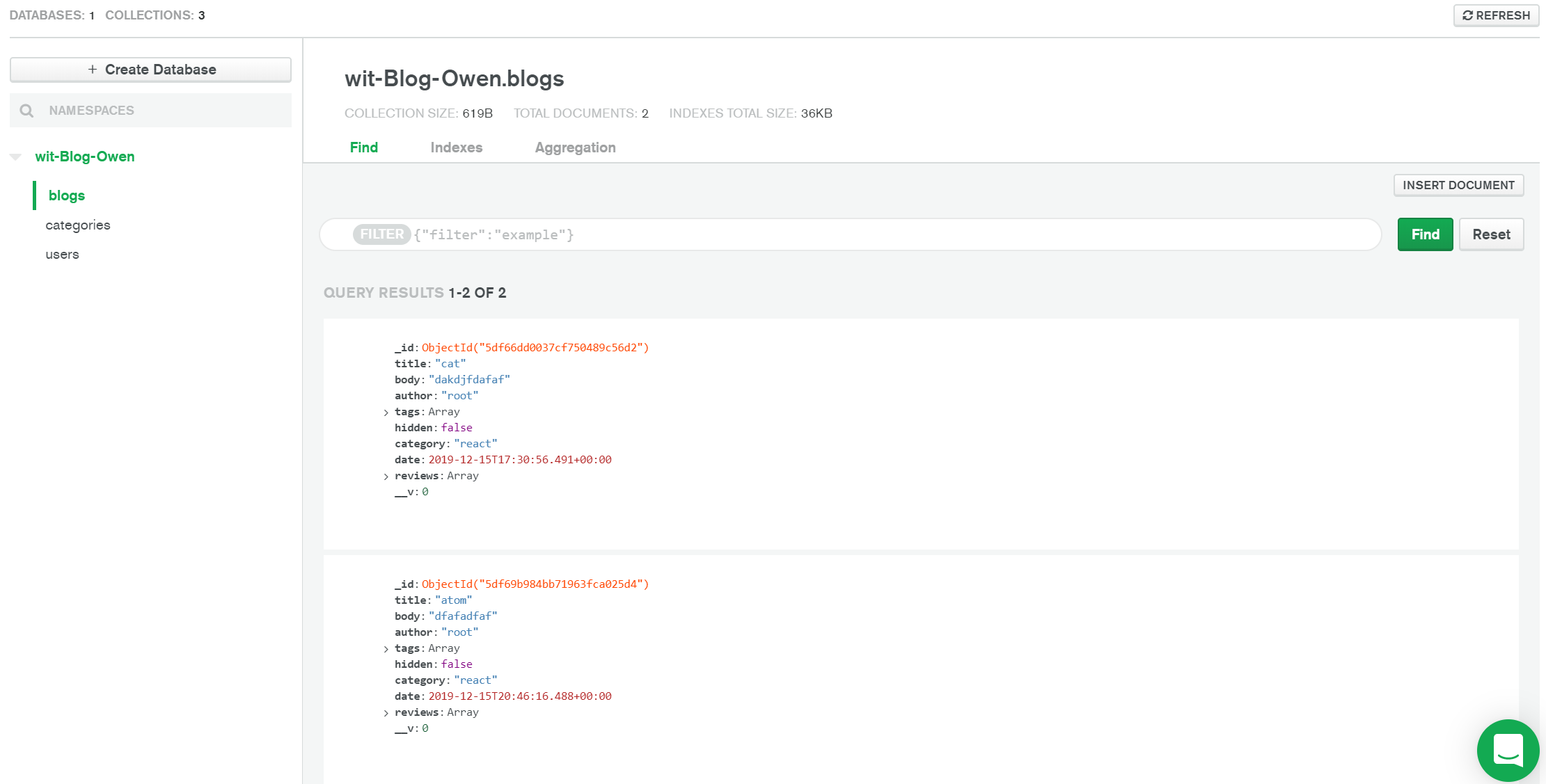
**a. User Database**



**b. Categories Database**



**c. Blogs Database**



***6. GIT Approach***

\* Back-end: <https://gitlab.com/owenego/node>

\* Front-end: <https://gitlab.com/owenego/blog-web>

***7. Software Test***

\* Back-end CI/CD Pipeline test:

<https://gitlab.com/owenego/node/pipelines>

\* Back-end Gitlab Pages:

<https://owenego.gitlab.io/node/>

\* Back-end Heroku-staging Pages:

https://owen-blog-staging.herokuapp.com/

<https://dashboard.heroku.com/apps/owen-blog-staging>

\* Back-end Heroku-production Pages

<https://dashboard.heroku.com/apps/owen-blog-production>

<https://owen-blog-production.herokuapp.com/>

\* Front-end Cypress test dashboard

<https://dashboard.cypress.io/projects/74c4nn/runs/14/specs>

***8. References***

a. https://segmentfault.com/a/1190000008629632

b. https://nodejs.org/zh-cn/docs/

c. https://expressjs.com/en/resources/frameworks.html

d. https://www.w3cschool.cn/mongodb/

e. https://www.runoob.com/nodejs/nodejs-npm.html

f. https://tutors-design.netlify.com/course/wit-wad-2-2019.netlify.com