Andrea Sanchez
Santiago Barrios
Alexandra Tabares
Bruno Montes

# Sprint 4 Documentation

## <u>**Detailed Work in Sprint 4:**</u>

**Front-End:**

Created and integrated a search bar into the “Cats” page that searches cats based on their name.

Created and set up routing for our “Cat Match Quiz” page so that the user is directed to the page directly from the home page via the “Find Your Match” button.

Quiz consists of HTML-type input of radio, which uses a type of multiple choice answer format to allow users to select only 1 option for what type of personality they would like their cat to be and lock in their response with the “Show My Partner” button.

The contact page has been updated to have a much cleaner look.

Added more cypress/unit tests for the new and already existing features added to the website.

What was ***<u>NOT</u>*** accomplished was an integrated connection for the front end to be able to display data such as the image of the cat, and the name of the cat, as well as some attributes. This was mainly due to poor timing and we hope to not repeat this issue in actual projects and in the industry.

**Back-End and API Documentation:**

*func getCats(w http.ResponseWriter, r *http.Request)*

The getCats function writes a method header of statusOK when the endpoint has been successfully reached and helps convert all of the list of cats into a JSON struct to be sent to the front end. The getCats function uses the GET method to portray all the cats on the screen.

*func getCat(w http.ResponseWriter, r *http.Request)*

This function is only slightly different from getCats in that it still uses the GET method, but it only returns a singular cat object after specifying the id of the cat. It also writes the method header of statusOK when the endpoint has been reached.

*func deleteCat(w http.ResponseWriter, r *http.Request)*

The deleteCats function writes a method header of statusNoContent when the endpoint has been successfully reached and finds the cat object in question by looking at its id and deleting it from the slice of cat objects. Afterwards it then encodes it as JSON to send the frontend a new updated list of cats. The deleteCats function uses the DELETE method to help get rid of the cat in question.

*func updateCat(w http.ResponseWriter, r *http.Request)*

The updateCats function writes a method header of statusOK when the endpoint has been successfully reached and finds the cat object by looking at its id and updating its content. After decoding the JSON and updating its content. It will put the cat object back into the list and then encode it to send to the frontend. The updateCat function uses the PUT method to help channel this process of updating the cat to the frontend.

*func createCat(w http.ResponseWriter, r *http.Request)*

The createCat function writes a method header of statusOK when the endpoint has been successfully reached and creates a new cat object to display to the frontend. It does this by decoding the information from the JSON and then writing it back to show to the frontend.

**Updated:**

Created the Get request function for the Cat Match Quiz *func GetCatQuiz(w http.ResponseWriter, r *http.Request)* based on the feature characteristic in the cat object for each cat. Therefore, the user picks the feature that they are interested in. Our function receives that input and then filters the cat slice and keeps the cats that have that feature and displays them back to the user.

Created the CreateUsers function that would route the “/contact” url to take in user data and convert that into a slice for database storage. Also created a user struct that would capture the JSON information from the frontend to ensure that the data is properly stored.

**Front-End Unit Tests:**

**OLD:**
Cypress Component Test - Forum Access and Interactivity for Contact Page

This unit tests for the ability to click on and access each forum field as a way to verify that the input text box is interactable. We want to be able to have the the functionality of filling out the forum automatically. - Passed (Partially because we want the filling feature by next sprint)

Cypress Component Test - Button Functionality for Home Component

This unit tests for click functionality for the “LOCAL cats” button on the home page for the website. - Passed

Cypress Component Test - Button Functionality for Cat Component

This unit tests for click functionality for the “Adopt” button on the home page for the website. - Passed

Cypress Component Test - Home

This unit tests for a valid routing when the user clicks on the “Home” button on the navigation bar on the webpage. - Failed (Now Passing in Video Demo)

Cypress Component Test - Shelters

This unit tests for a valid routing when the user clicks on the “Shelters” button on the navigation bar on the webpage. - Failed (Now Passing in Video Demo)

Cypress Component Test - Contact

This unit tests for a valid routing when the user clicks on the “Contact” button on the navigation bar on the webpage. - Passed

Cypress Component Test - About

This unit tests for a valid routing when the user clicks on the “About” button on the navigation bar on the webpage. - Passed

Cypress End-to-End Testing

This unit tests for a valid routing when the user clicks on the MAIN Page of website as well as other default end to end testing provided by Cypress. - Passed

**NEW:**

Quiz Selection and Submission Cypress Component testing - 11 test cases that perform the following: Select one of the 11 present bubble in options and clicks on the submit button at the bottom of the page. This allows us to determine the validity of the quiz's functionality as well as its routing with the backend.

**Back-End Unit Tests and functions:**

**OLD:**
*func TestInWebsiteMiamiDade(t *testing.T)*
*func TestInWebsiteLakeCounty(t *testing.T)*
*func TestInWebsitePeggy(t *testing.T)*
*func TestInWebsiteKeyWest(t *testing.T)*
*func TestInWebsiteMarathon(t *testing.T)*

The above five unit test functions test if the correct cats are being updated into the cat list. We created separate functions that convert the cat list into a string to be able to compare them in the unit tests. However, this can change because the shelters update their cats every day.

*func TestInWebMarathonAge(t *testing.T)*

This unit test function checks if the Age property of the cat object in Marathon Animal Shelter returns "In Website" since the HTML tags do not return the age of the cat for this shelter.

*func TestInWebLakeCountyBreed(t *testing.T)*

This unit test function checks if the Breed property of the cat object for Lake County Animal Shelter returns "In Website". This test is needed because the Lake County Animal Shelter does not display the breed of the animal on the website.

*func TestFeaturesMiamiDade(t *testing.T)*
*func TestFeaturesLakeCounty(t *testing.T)*
*func TestFeaturesPeggy(t *testing.T)*

The above three unit test functions test if the addition of the feature of personality for the matching quiz is correctly added to each cat in the shelters.

**We also were able to fix the following unit tests:**

*func TestGetCats(t *testing.T)*

This unit test creates a router and checks for a 200 ok status to see if it pings to the correct routing of getCats. The test also checks for an expected body response of getting all the cats.

*func TestGetCat(t *testing.T)*

This unit test creates a router and  checks for a 200 ok status to see if it pings to the correct routing of getCat. The test checks for an expected body response of a singular cat.

*func TestUpdateCat(t *testing.T)*

This unit test creates a router to help test for the updateCat method and compares if it connects by testing the connection status. The test then checks for an updated cat JSON struct.

*func TestCreateCat(t *testing.T)*

This unit test creates a router to help test for the createCat method and compares if it connects by testing the connection status. The test then checks to see if the cat struct has been created.

*func TestDeleteCat(t *testing.T)*

The unit test creates a router to help test for the deleteCat method and compares if it connects by checking the connection status, the status should be NoContent. If the correct status shows up, then the test has passed.

We were able to fix them because in the previous sprint, we had placeholders that were not the actual extracted cats from the shelters to be able to test if the functions worked. However, now we were able to hook up the actual extracted cats and display them when the tests are being ran.

**NEW**

func TestGetCatQuiz(t *testing.T)

This unit test creates a router that helps test the GetCatQuiz method which obtains the cats that contain the features the user has inputted from the quiz. Therefore, it checks if cats are obtained from the body response.

func TestCreateUser(t *testing.T)

This unit test creates a router that takes in a contact user information and checks if the information is successfully stored. If it is an error it says “handler returned wrong status code:” with the given code that the error generates.