

Unit Tests

INSTRUCTIONS: There should be details (can be in a separate file in the directory) provided by each team member about the module and the functional testing they have done. Each team member picks a module or module and lists the equivalence classes and the test cases selected to cover all equivalence classes.

(can remove above later)

Bryan:

Sidebar/Map

Map Page: Pulls and push reviews from/to database. The map to the right should update location as a location is searched/verified.

Testing:

1. Check if reviews are pulled from the database
2. Check if pins on the map represents reviews
3. Mouse click over the pinned locations to check if a review will pop up
4. Mouse drag on the map to see if map will move
5. Scroll on the map to see if map will zoom in/out
6. Search a new location to see if map will be updated (requires 2 clicks)
7. Click on review to see if a review form will show up
8. Submit a legit review. All textbox should be filled. Verify location twice and submit. Submission should be successful. Refresh page and verify if review has been added
9. Submit a review blank to see if alert box will pop up telling user to verify a location
10. In review, click verify twice without an input in "Location" text box and submit a blank form to see if submission fails
11. In review, click submit after filling in all input except location and verify location to check if submission will fail
12. Submit a review after filling all input with fake location (i.e random letters and number) to see if submission will fail
13. Search location first, then submit a review and verify a location with random numbers and letters. Fill out all fields. Submission will succeed. (known bug)
14. Submit a legit review and verify if location will show up on the sidebar and on the map after refreshing the page
15. Submit a legit review but change the address and click verify once. Submission will be successful, review will pop up in initial address (known bug)
16. Submit a legit review but change the address and click verify twice. Submission will be successful, review will pop up in new address
17. Submit a review with any textbox empty and check if submission fails

Anthony: I will be talking about the profile and the filter issues

In regards to the 'Profile' page and the display of user-submitted reviews:

Profile Page: Pulls data from the DB, filters reviews and displays only user-submitted reviews

Testing:

1. Logged in to see if 'Profile' appears on navigation bar
2. Log out to see if 'Profile' should disappear from the navigation bar
3. Click on 'Profile' to see if the profile page is navigated to properly
4. Resize window to see if any elements get 'bugged' out
5. Refresh page to see if elements on profile page is present
6. Click on 'Profile' to see if the username, email, and user-submitted reviews are shown
7. Navigate from 'Profile' to 'Map' and back again to ensure routing is correct
8. Submitted a review on 'Map' page, refreshed page and navigated to 'Profile' to ensure user-submitted review appears correctly on page
9. Refresh page to ensure elements are still shown after updating with a new review
10. Navigate from 'Profile' to 'Home' and back to ensure routing is correct

Vincent:

Logout button: Button on navbar, to logout of account

Testing:

1. When logged in, test that logout button works from Home page
2. When logged in, test that logout button works from Map page
3. When logged in, test that logout button works from Profile page
4. When logged in, test that logout button displays in upper right hand corner from Home page
5. When logged in, test that logout button displays in upper right hand corner from Map page
6. When logged in, test that logout button displays in upper right hand corner from Profile page
7. When logged out, test that the link to the Profile page does not display
8. When logged out, test that Logout button does not display
9. When logged out, test that the Login button displays
10. When logged out, test that the link to the Map page does not display
11. When logged out, test that the link to the Register page displays
12. When logged out, test that the Home page displays

Joey:

Home page/ Login/ Registration

Home Page: Pulls data from the DB and displays onto map with reviews

Testing:

1. Refresh the home page from user point
2. Logged out refresh of home page to see if it responds
3. Logged in refresh of home page to see if it responds

4. Clicked on pin to check if it pops up details
5. Minimized page and maximized page to test if it reloads properly
6. Resized page
7. Added new review and refreshed page
8. Scrolled out of view for Maps API
9. Scrolled into view for Maps API
10. Highlighted page to check if there are any bugs
11. Clicked on home icon to see if it brings you back

Login Page: User login page with 2 entries, username and password.

Testing:

1. Tested valid credentials
2. Testing invalid credentials
3. Tested valid credentials with caps
4. Tested invalid with caps
5. Tested username including email
6. Tested username including password
7. Tested username without password
8. Tested username with sparse spaces
9. Tested refresh of page
10. Refreshed page while logging in
11. Tested logging in while resizing
12. Tested minimizing and maximizing
13. Resized page to see if login bugged out
14. Refreshed page during resizing

Registration Page: User registration page that allows users to post reviews and add locations

Testing:

1. Tested if appears when logged in
2. Tested if you can access when logged in
3. Tested if you can access from profile
4. Tested access from Map
5. Tested access from home
6. Rescaled page to fit lower resolution.
7. Minimized page to check to see if it breaks
8. Maximised page to see if any changes
9. Resized page to see if it breaks
10. Refreshed from home
11. Tested if any blanks could be left Name
12. Tested if blanks could be left from password

13. Tested if blanks could be left from email
14. Tested if blanks could be left from username
15. Tested if uncompleted profile registration would pass
16. Tested if partially uncompleted profile registration would pass
17. Tested if only emails are accepted
18. Tested if any special characters are rejected
19. Tested submit button with blanks
20. Tested submit button with full registration

Kaitaku:

Backend: Database Connection

Setup DB and Table

Testing

1. Setup MongoDB Cloud DB (Mlab)
2. Test administrative permissions for entire team
3. Test creating table by manually inputting dummy data
4. Test creating multiple tables

Create

Testing

1. Tested insert manually using Mongoose's (MongoDB Framework) save() function
2. Initially tested POST request using Chrome extension (Postman) to check that it was making correct request with correct request body.
3. Wrote several request body with different entities (primary key, default, not null ...) to make sure it was appropriately inserted in the database.
4. Tested results of empty data getting inserted
5. Tested results of duplicate input for a unique field

Read

Testing

1. Tested manually by using Mongoose's find(), findAll(), etc function to read from database
2. Tested GET request using Chrome Extension to check that client was getting correct response.
3. Initially included a success/failure message in the response to log to the user's console the information that was returned
4. Tested invalid GET request
5. Tested request for non-existent database

Update

Testing

1. Tested update by using Mongoose's find(), findById() function to retrieve the correct data. Then manually updated the returned data and used Mongoose's save() function to insert the updated data back into the database.

2. Tested results of when every key:value gets deleted and inserted
3. Tested results of duplicate input for a unique field

Delete

Testing

1. Used Mongoose's remove() function to delete entries from database
2. After deletion, checked the database to make sure it was deleted
3. Tested deletion of non-existing data