Test Results

Team: 17-03

Create/Login/Logout

Use Case	Results
Creating a user Account	**User 1**: Creates an account with valid username and password. **User 2:** Creates an account, is taken back because he has to include a special character in his password when creating. Fails first attempt and succeeds second attempt after putting in a valid password. **NOTE:** A valid password should be at least 8 characters long, should have one lowercase character, one uppercase character, one numeric digit between 0 and 9, and one special character. Special character must be one of this (~ ! # * $ @ _ % + = . , & ( ) { } | ; : < > ?) **Conclusion**: Creating an account was pretty self-explanatory and the users were able to create an account.
Entering Username and Password	**User 1:** User enters username and a valid password clicks and has no trouble logging into the home page of his account. **User 2:** User enters username and a wrong password and is let known that his username/password is incorrect. Logs in again and successfully is able to. **Conclusion:** Behavior is consistent and results are almost as expected. It took 2 tries for one person to succeed in completing this but that was because of human error.
Logging out of account	**User 1:** User is able to click on the logout button and log out as expected. **User 2:** User is able to click on the logout button and log out as expected. **Conclusion:** Behavior is consistent and results are as expected.

Discover

Use Case	Results
Navigating to the Discover page displays recipes from API	**User 1**: User clicked Discover, and the page loaded a list of recipes. **User 2**: User clicked Discover, and the page loaded a list of recipes. **Conclusion**: Behavior is consistent and results are as expected.
Entering input into search returns recipes matching the criteria	**User 1:** User types in Potato in the search field, and the page reloads showing recipes with potato. **User 2:** User types in Pasta, and the page reloads displaying recipes containing pasta. **Conclusion:** Behavior is consistent and results are as expected, with the search results correctly reflecting the input.
Add recipes to library on button click	**User 1:** User clicked "Add to Library" next to chosen recipe and the success message displayed. User navigated to Library (as requested) and added recipe was displayed. **User 2:** User clicked "Add to Library" next to a recipe, continued browsing, and clicked "Add to Library" on a second recipe. Success message displayed for both. User navigated to Library (as requested) and added recipes were displayed. **Conclusion**: Behavior is consistent and results were as expected, with recipes confirmed to have been added to Library.

Library

Use Case	Results
Recipes associated with the user populate the page	**User 1:** User navigated immediately to the library page from the navigation bar (user was using the test account-> recipes are automatically inserted into the library). All recipes in the library populated the page. **User 2:** User created and logged into a new account. Immediately, the user navigated to the library page. There was no data to load, as expected. The user navigated to the home

	page and added a recipe to the library via the add meal modal. Then, the user navigated to the discovery page and hit the 'Add to Library' button. Afterwards, the user navigated to the library page and both recipes were populated on the library page. **Conclusion:** Behavior is consistent and the program performed as expected. In the case where the user 1 logged into an account with pre-existing data, all data was populated. When user 2 created a new account and added recipes to the library, those new recipes populated on the library page, meaning they were stored correctly in the database.
User can search for specific recipes	**User 1:** The users library included default recipes added to the library, as seen in the use case above. When searching for one of the recipes using the top left search bar, the user had caps lock on. After the search returned nothing, the user realized the search was case sensitive and adjusted their search. This time around, the page was populated with recipes with 'ee' somewhere in the name. **User 2:** The user's library was populated with the data discussed above. The user typed 'a' into the search box, and one recipe displayed on the library page (one recipe had character 'a' in the recipe name). After, the user navigated to the home page and then back to the library page. However, the page only displayed the recipe including the character 'a'. **Conclusion:** The behavior for the search functionality worked, but had some unintended mechanics. The search/sort should reset when the user navigates to the library page from the navigation bar.
User can sort recipes by nutrient	**User 1:** User 1 sorted their recipes by each nutrient by clicking the small triangles by every nutrient name- calories, protein, carbs, and fat. The nutrients were sorted in ascending, descending, ascending, and ascending order respectively. **User 2:** User 2 sorted their recipes by protein and carbs, and then navigated to the home page. Upon returning to the library via the navbar, the sort was refreshed. The user sorted by protein again and got the same results. **Conclusion:** Sorting behavior was consistent with expectations and was simple and easy to use. Returning to the library page via the navbar resets whatever sorting is applied to the data set.
User can search for specific recipes, then sort by nutrient	**User 1:** User 1 repeated their search from the use case above, and the library populated as expected. When the user sorted by protein, the Library page displayed the recipes matching the

search criteria sorted by their protein content in descending order. The user then sorted by fat, and the same recipes were resorted by fat content in ascending order.

User 2: User 2 only had two recipes, so they returned to home to add one more to the library. User 2 sorted the recipes initially by protein, and then searched for two of the recipes. The page rendered unsorted, but then the user was able to sort by protein.

Conclusion: While sorting didn't apply after the library was searched, this behavior was to be expected. In other cases, the user was able to search for recipes and then sort them seamlessly.

Profile

Use Case	Results
Profile displays current user's info	**User 1:** User (in test account) navigates to profile page from menu bar. Test data for 'about me' and preferences is displayed. **User 2:** User (in new account) navigates to profile page from menu bar. Fields state they are empty. Appreciates the instructions beneath fields. **Conclusion:** Behavior is consistent, profile page displays correctly, descriptions for entry criteria are useful.
Profile allows user to make edits	**User 1:** User (in test account) clicks Edit. Is unsure if click worked, about to click again and notes the form has changed. User modifies profile, and clicks Save. Upon request, navigates away and returns; modifications are present. **User 2:** User (in their new account) clicks Edit. Inputs info into the fields and clicks Save. Upon request, navigates away and returns; entries are present. **Conclusion:** Behavior is consistent, profile is updated successfully and changes are correctly stored and reflected.

Homepage (with Add/Edit/Delete Meal)

Use Case	Results
Add Meal on home page, user entry	**User 1:** User clicks add; wants to know what their options are, and after being told about the Library, closes modal to go look. User is not interested in library options and decides to return Home to add Waffles and Strawberries manually. Asks if she can put 0 calories, decides on -1. Fills in rest of the fields based on assumption that 1 serving is equal to 3 waffles. "I'm gonna save this recipe because it's good," she says, and checks the box and saves. Is disappointed in the default image. **User 2:** User clicks add, enters 'Doom' with 10 servings. Enters 'no' for calories.Searches for an image to add. Tries to save the meal, but form will not submit due to invalid entry 'no.' Adjusts calories to number and successfully adds. **Conclusions:** Behavior is mostly consistent with the use case, no apparent obstacles in navigation; ability to add negative values is a problem. Inability to add text to number fields is probably for the best. Unreasonable servings and nutrition content outside of developer jurisdiction. **Changes:** Adjusted fields to not accept negative values.
Navigate between days on home page	**User 1:** User wants to go to tomorrow, asks if she can do that on the Calendar page and clicks Calendar without waiting for response, then selects tomorrow from Calendar. **User 2:** Clicks the forward arrow to go to tomorrow. **Conclusions:** Calendar may position itself as the more obvious choice for date navigation, but it's still good to have the option on home. Possibly a problem with arrow display; second user had no issues, so tentatively should be fine. **Changes:** Arrow buttons were adjusted to hopefully be more prominent.
Add meal on home page, from library	**User 1:** Clicks add meal and adds 'Garlicky Pasta,' which she won't eat because she already had too many waffles, but is apparently okay, because she can't eat garlic anyway. ("Good thing I have a dog," she says.) Wants to know why she's eating it at midnight and if she can change that. Decides to add meal she can actually eat, and adds Eastern European Lentil. Decides to then add scrambled eggs and bacon; wants an appetizing picture, so

	goes to find an image. Adds url and clicks save. Meal appears in log.
	User 2: Clicks add, types in I, selects one of the options in autocomplete, and clicks save. Meal appears in log.
	Conclusions: Behavior is consistent and meal adding from library seems functional, but defaulting to midnight for meals added to previous or later days seems to be a problem.
	Changes: Time is now user-entry and editable.
Edit servings on home page	**User 1:** Clicks Edit, changes servings to 2 and saves. Edit reflected in log.
	User 2: Clicks Edit, changes servings to 200 and saves. Edit reflected in log.
	Conclusions: No issues/deviation (restricting servings number seems unnecessary and overbearing.)
Delete meal on home page	**User 1:** Clicks delete under the meal she wants to delete; meal is removed.
	User 2: Clicks delete under a random meal; meal is removed.
	Conclusions: No issues/deviation.

Calendar

Use Case	Results
Navigate between weeks on Calendar page	**User 1:** Promptly clicks back arrow to go back a week. Previous week's info is displayed.
	User 2: Promptly clicks forward arrow to go forward a week. Blank cards of future week are displayed. I try to thank her for her time, but she says 'Hold on, it's a Thursday, I need to eat food again,' and clicks Thursday to add another meal.
	Conclusions: No issues with week navigation.
View Meal on Calendar page	**User 1:** Promptly clicks the name of a meal, successfully views info.
	User 2: Clicks on the waffles and strawberries added by first user, successfully views; is shocked -1 calories was an option.

	Tries to adjust it and is able, though there is no submit button to save changes. **Conclusions**: No issues with modal popping up and showing correct info, but fields should be inactive on View. **Changes:** Fixed fields to be inactive.
Jump to day from Calendar page	**User 1:** Successfully did so in another use case. **User 2:** Alarmed by empty spaces of future days; I invite her to 'go to' one of them and add a meal. She clicks Thursday, is taken to the day's log, and adds a new meal, represented by a flock of flamingos, saving it to the library. **Conclusions:** No issues/deviation.