

1. Create 1 document per team that describes how at least three features within your finished product will be tested.
2. The test plans should include specific test cases (user acceptance test cases) that describe the data and the user activity that will be executed in order to verify proper functionality of the feature.
3. The test plans should include a description of the test data that will be used to test the feature.
4. The test plans should include a description of the test environment that will be used to test the feature.
5. The test plans should include a description of the test results that will be used to test the feature.
6. The test plan should include information about the user acceptance testers. Please note that the deployed application can only be tested on the CU Boulder campus. So make sure to select your testers accordingly.

Features to be tested:

- Register / login page
- Ability to pull recipe from database
- Ability to pull pricing from Kroger API

Specific Test Cases, Test Data, and Users:

- Register / login page
 - For users who would like to create an account or log in to an existing account, the user acceptance test case would be a user creating the account and when revisiting the website, being able to log back in.
 - The test data the user is able to provide their desired email, username, and password, which can then be confirmed to be inserted into the users table, and which can then be used to log in. The test data used will be sample users, with sample emails, usernames, and passwords, which can be inserted into our backend users table. From here, the user can then log in using these sample users.

- The test result for register is once you register, your information should be saved with the data. With that the next time when you login, your email and password is saved and you are able to login. Before that, if you have not created an account, you should not be able to login.
- User acceptance testers will be students from CU Boulder with no previous knowledge of our application, who can then try different inputs for our fields.
- Ability to pull recipe from database
 - The user should be able to interact with the application and easily request a recipe on the recipes page/API.
 - Tests include successful interactions with our API in retrieving recipes and ingredients.
 - The test data used is example recipes added to the recipes table, which includes the recipe name, recipe description, and recipe image, as well as a recipe to ingredients table.
 - Users should be able to interact with the application on their own individual laptop.
 - Test with CU Boulder Students who are from a variety of backgrounds
 - Users should be able to retrieve recipe data associated with their individual registered account after using log in.
 - The test result is that the user is able to see the results of a specific recipe as according to their request.
 - A user's activity of any interaction with data associated with recipes and ingredients through the UI should be successful.
- Ability to pull pricing from API
 - Making sure we can find ingredient pricing and return it to the user for each recipe.
 - Use example recipe data we create or test users upload.
 - Sample the app to students on campus who are willing to participate and from different backgrounds over the course of a week.

- Expect to see accurate pricing in line with King Soopers website and in store prices
- Test with CU Boulder Students who are from a variety of backgrounds
- The test result is that the user is able to see the live price of a specific ingredient according to King Soopers.
- Authentication for API key should be functional (no errors on user's end).
- Testing should be on a diverse array of CU Boulder students who are located near a King Soopers.