

System and Unit Test Report

Project Name: Boba Tracker

Team Name: Boba Peeps

Date: 06/01/2021

System Test Scenarios

Sprint 2:

User story 3: As a user, I want to be able to create an account to maintain my privacy through a username and password.

1. Go to bobatracker.com
2. Select 'Create an account!'
3. Type:
 - a. Name = Frank Ocean
 - b. Email = frankocean@gmail.com
 - c. Password = frankocean1
 - d. Confirm Password =frankocean1
4. Press 'Register' button
5. This will login and redirect to the 'Dashboard' page where the user can see a message saying "Welcome Testing2!"
6. Select 'Logout' on the navigation bar at the top of the page.

Sprint 1:

User story 1: As a user, I want to be able to input how much I spend on each drink and view it.

User story 2: As a user, I want to view my information in a chart.

1. Go to bobatracker.com
2. Login into account, type:
 - a. Email = frankocean@gmail.com
 - b. Password = frankocean1
3. Press 'Sign in' button
4. Select 'New Drink' on navigation bar at the top
5. Add new drink
 - a. Input date as: 6/01/2021
 - b. Input drink name as: Thai
 - c. Input price as: 5.00
 - d. Input sweetness: 0.75
6. User should see a message saying 'New drink added!'
7. Select 'Dashboard' on navigation bar at the top
8. User should be able to see
 - a. Drinks bought this week as: 1
 - b. Money spent as: 5.00
 - c. Input sweetness as: 0.75
9. The user should be able to see the chart on the left with y-axis between 6 and 5 and x_axis on Tuesday.
10. Select 'Logout' on the navigation bar at the top of the page.

Sprint 3:

User story 2: As a user, I want my data stored securely and independently in a database (13 Story Points)

User story 3: As a user, I want to be able to view my purchase history.

1. Go to bobatracker.com
2. Login into account, type:
 - a. Email = frankocean@gmail.com
 - b. Password = frankocean1
3. Press 'Sign in' button
4. This will login and redirect to the 'Dashboard' page where the user can see a message saying "Welcome Frank Ocean!"
5. User should be able to see
 - a. Drinks bought this week as: 1
 - b. Money spent as: 5.00
 - c. Input sweetness as: 0.75
6. Select 'Logout' on navigation bar at the top
7. User should be redirected to the 'Login' page
8. Select 'Create an account!'
9. Type:
 - a. Name = DifferentUser
 - b. Email = differentuser@gmail.com
 - c. Password = differentuser1
 - d. Confirm Password = differentuser1
10. Press 'Register' button
11. This will login and redirect to the 'Dashboard' page where the user can see a message saying "Welcome DifferentUser!" and will not be able to see anyone else's data and since it is a new user, should be able to see:
 - a. Drinks bought this week as: 0
 - b. Money spent as: 0
 - c. Input sweetness as: 0
12. Select New Drink on navigation bar at the top
13. Add new drink
 - a. Input date as 6/01/2021
 - b. Input drink name as: Taro
 - c. Input price as: 4.50
 - d. Input sweetness as: 1
14. User should see a message saying 'New drink added!'
15. Select 'History' on navigation bar at the top
16. Under 'Past Purchases' user should a box with the drink information just added in the order:
 - a. 2021-06-01
 - b. Taro
 - c. Price: \$4.5
 - d. Sweetness level: 1
 - e. An 'Edit' button and a 'Delete' button.
17. Select 'Logout' on the navigation bar at the top of the page.

Sprint 4:

User story 1: As a user, I want to view the sugar content in a chart.

User story 2: As a user I want to be able to edit previous entries in case I make a mistake.

1. Go to bobatracker.com
2. Login into account, type:
 - a. Email = differentuser@gmail.com
 - b. Password = differentuser1
3. Press 'Sign in' button
4. Select 'History' on the navigation bar at the top
5. Under 'Past Purchases' user should see a box containing a previous drink entry with the following information:
 - a. 2021-06-01
 - b. Taro
 - c. Price: 4.50
 - d. Sweetness level: 1
 - e. An 'edit' button and a 'delete' button.
6. Select the 'edit' button on the box.
7. Users should be redirected to a page with a form. The form will have the input values prefilled with the drink information they selected to edit.
8. Change sweetness level to 0.25
9. Select 'Save'
10. A message box will appear with the message "Successfully added!"
11. Select 'close'
12. Select 'History' on navigation bar at the top
13. Under 'Past Purchases' sweetness level should now be 0.25
14. Select 'Dashboard' on the navigation bar at the top
15. The user should be able to see a chart on the right with a data point (0.25, 1.0), showing 1 drink with 25% sweetness.

UNIT TEST

The unit tests focus on testing the backend functionalities to make sure they work correctly for the frontend to use. To run these unit tests, navigate to the backend then run **npm test** or **npm run test**

GET Invalid URL:

When a invalid url is entered, the return code should be 404

GET All:

Test that it gets all the bobas in the database. Check to see if all the fields are there such as purchase_date, drinkname, sweetness

GET All for one user:

Tests to see that it can get all drinks for one user. So will check for purchase_date, drinkname, sweetness, price, and email matches the one entered into pathname

GET One Boba:

Hardcoded a boba id into the url path and checked to see it got the correct drink. Checked to see if we got the expected email, drink name, price, and sweetness level

POST Boba:

Checks to see that it can correctly POST/upload a new boba drink for the specified email. Create a object with all the specified fields : purchase_date, drinkname, sweetness, price, and email, and make sure the return status code is 201

DELETE Boba:

Deleted a boba using a random id. Make sure we got the expected return status code with is 200. Then went to expressAPI to manually check that boba order got deleted, because can't think of another way to make sure it got deleted.