Foodie

Spring 2018

Team members:
Guocheng Wei, Jiayi Kou, Yuchen Zhang, Zhaoya Sun

Design Inspection

Code Inspection

Unit Testing Log

Design Inspection

Product	**Foodie**
Date	02/08/18
Author	The Foodie Team
Moderator	Yuchen Zhang
Inspectors	Zhaoya Sun, Jiayi Kou, Guocheng Wei, Yuchen Zhang
Recorder	Zhaoya Sun, Jiayi Kou

Defect #	**Description**	**Severity**	**How corrected**
1	Ingredient name is not enforced to be a valid ingredient. Valid here means to be able to eat. User can type in anything even if they are not normal ingredients, such as stone or pen.	2	Users will be notified of the possible ingredient names as they type in the first few letters. [Yet to implemented]
2	Currently we only retrieve 1,000 recipes to store in our database through API call for users to search.	2	If users required more recipes in the future, our project owner will expand our database storage and retrieve more recipes.
3	Users might be confused of two search pages we designed, one is based on the ingredients user has to get recipes and the other is to search recipes with specific	1	We will provide user with a help portal to guide them on how these two search pages work.

	keywords.		
4	User doesn't have username in his/her profile. Only email is taken as user's unique id. Email is relatively longer than username and more private.	1	We could add username as another attribute of user to make profile more personalized.[yet to be implemented]
5	User might be confused about seperated user profile and enter ingredient page	2	We put the enter ingredient page in the user profile page so after user signs in, he/she will be redirected to the profile page to add ingredients. User can also navigate back to the user profile to modify his/her ingredients.
6	User won't be notified if he/she hit the add button without enter anything in the ingredient field	3	We add an alert message to notify the user that ingredient field can't be null.

Code Inspection

Code inspection revealed the following defects:

Product	**Foodie**		
Date	02/08/18		
Author	The Foodie Team		
Defect #	**Description**	**Severity**	**How corrected**
1	Use 'isTagFieldNull' boolean state variable in Profile component when it	3	Use (tagField === '') instead to check tagField input

	is not needed.		
2	Gives ‘user not defined’ error when Meteor.users is called. Component rendering faster than meteor user async calls	1	Display a ‘Loading’ state before the async call returns and check if user is undefined before using user object.
3	When retrieving json object from API, we used object1.object2	1	Use object1[“object2”] instead to retrieve json objects from Edamam API.
4	The SCSS coding style looks the same as CSS syntax	3	Modify the coding style to take the advantage of SCSS

Unit Testing Report

The system relies on the following modules:

- Page Layout
  - Header
  - Main content
  - Footer
- Header
  - User log in
  - User sign up
  - Change password
  - User log out
  - Page navigation
- Home
- Profile
  - Edit avatar picture
  - Delete account
  - Enter/modify ingredients
- Saved ingredients
- Recipe browser
  - All recipes
  - Recipes based on ingredients

- Detailed recipe

Header interface

The header interface in Foodie is the key to page navigation, user log in/log out behavior and account creation. We used a navigation drop down style to organize user behaviors in one button. Router is done using react-router-dom package and the behaviors are changing as user logs in and out. The testing

Product	**Foodie**		
Module	Header		
Date	1/30/2018		
Author	Guocheng Wei		
Defect #	**Description**	**Severity**	**How corrected**
1	When creating a user, there is an server internal error reported by MongoDB. Since Foodie uses the SimpleSchema to restrict the “users” collection document style, when creating a user, there is no default value to those mandatory fields.	1	Assign default value to the each of mandatory fields of “users” schema.
2	When creating a user, besides the “at least 6 characters”, the password does not contain at least one of the two of three requirements: number, uppercase letter or lowercase letter.	1	Use RegExp in to test the validation of the password.
3	When using the facebook oauth to login, facebook returns a 403 access error	1	Since Foodie is in the developing mode, which means the request is in the “http” instead of “https”, only known users can use this service. We add all

			developers into the white list in the Facebook developer console of Foodie app.
4	When users login, the app did not redirect them from “Home” page into the “Search” page	1	Add a authorization step in the react Router. Use ”Meteor.userId()” to check if a user login. Then if so, redirects the user to the “Search” page by using <Redirect> in the react-router-dom
5	When a user logs in, Foodie does not rerender the header to show all the available pages links	1	Use TrackerReact to check if the Meteor.userId() exists. If so, automatically re-renders only the head in the react dom.
6	When a user logs out, Foodie redirects to the “Home” page.	1	Based on the Defect 4, if ”Meteor.userId()” returns undefined or null, redirects the user to the “Hom” page by using <Redirect> in the react-router-dom

Home interface

The home page is the first page the users see when they find our website. The user will be redirected to create an account or sign in and they won’t be able to see other modules before they sign in.

Product	**Foodie**		
Module	Home		
Date	02/05/2018		
Author	Zhaoya Sun		
Defect #	**Description**	**Severity**	**How corrected**

1	The background image of homepage cannot be set up appropriately with the framework of ant design. The image always appears to be within a border of white and cannot cover the whole page under the header.	2	Modified background image's value of content margin and content padding to 0, making it reach to all edges of homepage.
2	The header doesn't show user's detailed modules after the user log in or sign up.	1	Be careful to make the router redirect to each user's specific header content.
3	The opacity of images and text on the homepage is not separated. The change to opacity of images would also influence the opacity of text.	2	Create new blocks for text out of images so that the style of them can be modified separately.

Profile interface

The profile interface displays the user's basic info and user will be able to enter and modify their ingredients from this page. User can also manage their account from here. The testing is done using javascript testing framework mocha.

Product	**Foodie**		
Module	Profile		
Date	2/1/2018		
Author	Jiayi Kou		
Defect #	**Description**	**Severity**	**How corrected**
1	Avatar picture not uploading. Inspection gives no errors and there is nothing stored in the database.	1	Change ant design uploading avatar framework to cfs:file-collection framework

2	Avatar picture uploaded but not appearing instantly	2	Use TrackerReact to make sure the component is rendering as the content is changing
3	Upload pdf files instead of picture files in avatar. The backend receives errors but nothing is happening in the front end.	3	Add ant design message to give the user an alert message when they upload wrong file type.
4	Email information is not in the center. Display content is not interactive and adjusted to the browser size.	3	Use ant design Card type to organize display and use display:inline-block to make sure everything stays in proportion.
5	Delete account is not functioning. Meteor.users.remove() failed because it prohibits such behavior without specification.	1	Add remove function in the users.collection schema.
6	Ingredient tags cannot be saved after hitting the save button. Error message: undefined 'profile.ingredients'	1	Define user schema in user.collection.js file where all user information fields are defined under profile schema.

Recipe browser interface

The recipe browser interface is divided into two parts: one where user can search through all recipes and another where user will see all the recipes generated based on the user's ingredients. Once the user clicks on each recipe, they will be redirected to the specific detailed recipe page.

Product	**Foodie**
Module	Recipe browser

Date	1/31/2018		
Author	Yuchen Zhang		
Defect #	**Description**	**Severity**	**How corrected**
1	Nothing is shown in the table when the page is loaded at the beginning	3	Add async call to the API before the the table component is rendered
2	The button group used for search mode direction does not change color if clicked /selected	2	Add a state variable to store the activated button and conditionally apply the style based on which button is clicked
3	Clicking the “more” button will replace the existent search result	2	Use es6 spread operator to append the newly requested results to the existing result array

Detailed recipe interface

The detailed recipe interface displays information from the Edamam API of each recipes including ingredients, nutrition labels, image and a link to the recipe webpage. The data retrieval is tested using mocha testing framework.

Product	**Foodie**		
Module	Detailed recipe		
Date	2/4/2018		
Author	Jiayi Kou		
Defect #	**Description**	**Severity**	**How corrected**
1	Recipe image doesn’t fit into the display content	2	Use css and set width, height, resolution and

			border size of the image.
2	Edamam API not responding from ‘GET’ call	1	Use axios framework to get Edamam API and put the ‘GET’ call inside ‘componentDidMount()’ method which waits for the async API call to call back
3	Iterating through health labels not working and labels are not displaying.	1	Use react array.map() to iterate through the labels and display them in Tags