

# Foodie

Spring 2018

**Team members:**

Guocheng Wei, Jiayi Kou, Yuchen Zhang, Zhaoya Sun

Design Inspection

Code Inspection

Unit Testing Log

Sprint 2

## Design Inspection

<b>Product</b>	Foodie		
<b>Date</b>	03/03/18		
<b>Author</b>	The Foodie Team		
<b>Moderator</b>	Yuchen Zhang		
<b>Inspectors</b>	Zhaoya Sun, Jiayi Kou, Guocheng Wei, Yuchen Zhang		
<b>Recorder</b>	Zhaoya Sun, Jiayi Kou		
<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How corrected</b>
1	User cannot receive a verify email to notify the user that they create an account in Foodie	1	Send an email when a new account is created.
2	User cannot go back edit/delete their comments if they typed wrong words or wanted to delete their comments.	2	If user wants to go back edit/delete their comments, there will be edit and delete buttons provided.
3	User cannot go back edit/delete their replies either if they found something wrong with their words	2	User will be provided with edit/delete buttons to do edit and delete.
4	User can like the recipe multiple times	1	In the client side, notify the user they already liked the recipe. In the server side, do not increase the like counts if the user already liked the recipe
5	User can save the recipe	1	In the client side, notify the

	multiple times		user they already saved the recipe. In the server, do not add this recipe info again into the Users collection
6	When the first time visit a detail recipe page without a doccollection in the database, there will be a render error since no result return from the server.	1	Use a sync call instead of async http request to get the data. Then, insert the api data with default value of likeCounts and viewCounts into the Recipes collection

## Code Inspection

Code inspection revealed the following defects:

<b>Product</b>	<b>Foodie</b>		
<b>Date</b>	03/03/18		
<b>Author</b>	The Foodie Team		
<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How corrected</b>
1	User may not be able to receive the email when he/she creates a new account or changes password because we have local host link in our email address	2	The issue will be fixed after we publish our website. Local host link will be substituted by a valid email address.
2	Use map instead of forEach when iterating the savedRecipe array in users.methods.js	3	Use forEach instead of map. Since we didn't need to create a new array, we chose to use foreach instead of map.

3	Use Meteor.callPromise instead of Meteor.call when saving a user's favorite recipe.	2	Since Meteor.call will return an error or success message and Meteor.callPromise will return callback, we chose to use Meteor.call. We didn't need any callback from saving recipe.
4	this.props returns undefined when passing recipe id from Recipe component in saveRecipe function	1	Add bind(this) to saveRecipe so saveRecipe's scope is on the Saved component.

## Unit Testing Report

The system relies on the following modules:

- Page Layout (Completed in sprint 1)
  - Header
  - Main content
  - Footer
- Home (Completed in sprint 1)
- Profile
  - Email Notification
- Recipe browser
  - All recipes
  - Recipes based on ingredients
- Detailed recipe
  - Like recipe
  - Save recipe
  - View counts
- Saved recipe

### ***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.

<b>Product</b>	<b>Foodie</b>		
<b>Module</b>	Profile		
<b>Date</b>	03/03/18		
<b>Author</b>	Guocheng Wei		
<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How corrected</b>
1	User cannot receive a verify email to notify the user that they create an account in Foodie	1	Add send Email functionality. Also, listen to the database, if a new user has been created, send a special verify email to the user's email.
2	Tags added were not directly saved into database, which might potential cause inconsistency.	2	Removed save button in the process of adding ingredients. The ingredients will be added to database once add button is pressed.

### ***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.

<b>Product</b>	<b>Foodie</b>		
<b>Module</b>	Recipe browser		
<b>Date</b>	03/03/18		
<b>Author</b>	Yuchen Zhang		
<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How corrected</b>
1	API request issue when adding diet	1	Have not fixed yet

	and health category for filtered searching		
--	--	--	--

### ***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.

<b>Product</b>	<b>Foodie</b>		
<b>Module</b>	Detailed recipe		
<b>Date</b>	03/03/18		
<b>Author</b>	Jiayi Kou, Yuchen Zhang		
<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How corrected</b>
1	Component was not testing-friendly	2	Breaking down one component into several ones
2	Nutrient table were unable to render	1	Check the data props for the table component first and filtered out invalid data.
3	Like and save buttons still have a border when hovering over them	3	Not yet corrected
4	Save button can save a certain recipe multiple times	2	Use "addToSet" instead of "push" when adding saved recipe to a user object. It avoids duplicates and it won't add a recipe already in the list.

<b>Product</b>	<b>Foodie</b>
----------------	---------------

<b>Module</b>	Saved Recipe		
<b>Date</b>	03/04/2018		
<b>Author</b>	Jiayi Kou		
<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How corrected</b>
1	Use RecipeTable component caused componentNotFound error	1	Pass in data attribute when using RecipeTable component.
2	RecipeTable data won't display and return Null property error	1	Get api_data attribute from recipe object instead of using recipe object directly.
3	Get recipe data on the front end. It's not a safe practice to use get and set methods on the front end.	2	We moved the get recipe data function to the backend. Use Meteor.methods to construct backend functions and use Meteor.call/callPromise to utilize the function in the front end





