



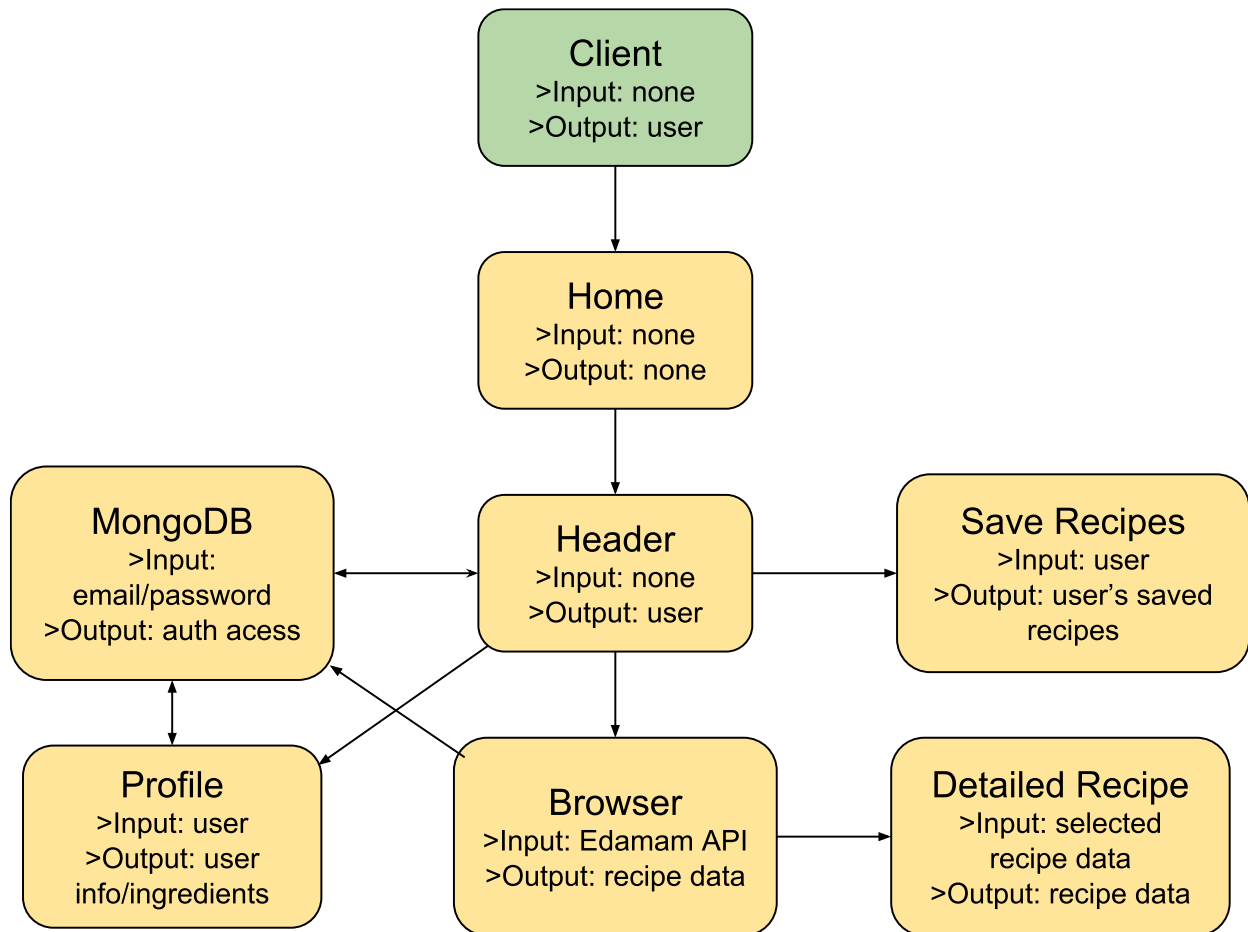
Spring 2018

**Team members:**

Guocheng Wei, Jiayi Kou, Yuchen Zhang, Zhaoya Sun

Regression  
And  
Incremental Testing  
Log  
Sprint 2

## Classification of Components



## Incremental Testing Techniques

We decided to use the top-down testing approach because testing the higher levels of our application will reveal if there are any bugs at the lower levels. In addition, our application is very straightforward, so it would be relatively easy to root cause bugs. This goes best with our

design approach of developing the core modules first and then adding and refining the lower levels.

## Incremental and Regression Testing

### Module: Login

Product	Foodie		
Module Tested	Login		
<div>Login</div>			
Incremental Testing			
Defect #	Description	Severity	How corrected
1	When creating an account, users can bypass the client side password verification code and send a insecure password to the server.	1	Insert a similar piece of code to check the security of the password in the server before creating an account.
Regression Testing			
Defect #	Description	Severity	How corrected
	No defects found		

### Module: Saved Recipe

Product	Foodie
Module	Saved Recipe

Tested			
<div>Saved Recipe</div>			
Incremental Testing			
Defect #	Description	Severity	How corrected
1	Data is undefined when RecipeTable component is called.	1	Use a state variable data to record data and check if it is undefined before calling RecipeTable component
2	Like and view counts are not shown in saved recipe page	3	Add likecount and viewcount to customColumn of RecipeTable
Regression Testing			
Defect #	Description	Severity	How corrected
1	Saved recipe component have to make several API calls to get the saved recipes' details	2	We created a backend to store user and recipe information when they were visited. Now saved recipe component only needs to call our own backend to get recipe information.

## Module: Profile Page

Product	Foodie		
Module Tested	Profile		
<div>Profile</div>			
Incremental Testing			
Defect #	Description	Severity	How corrected
1	The rendering of users' ingredients fails when switching from another module to the Profile module.	1	It was caused by using createContainer while not giving the tags state a value when the page is mounted. The data of users was not retrieved before the page start mounting. So instead of using createContainer, assign all the data to the states when the page is mounted.
Regression Testing			
Defect #	Description	Severity	How corrected
	No defects found		

## Module: Recipe Browser

Product	Foodie		
Module Tested	Recipe Browser		
<div>Recipe Browser</div>			
Incremental Testing			
Defect #	Description	Severity	How corrected
	No defects found		
Regression Testing			
Defect #	Description	Severity	How corrected
	No defects found		

### Module: Detailed Recipe Page

Product	Foodie		
Module Tested	Detailed Recipe		
<div>Detailed Recipe</div>			
Incremental Testing			
Defect #	Description	Severity	How corrected
1	Comment section doesn't	2	Since we don't have

	have user email when user is posting a comment		username for each user, we simply changed username to user email in the comment section so user's email is shown when he/she is posting a comment.
2	Comment section passes api_id of each recipe from Edamam	3	Use doc_id, which is our own recipe id in the backend. This ensures the security and stability of our comment section.
<b>Regression Testing</b>			
<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How corrected</b>
1	Like button doesn't have a special effect to let the user know he has clicked on it	2	Change css style of the like button to fill the heart after the user has clicked.
2	Logic of missing ingredients is wrong. When user has two ingredients, it appears that nothing was needed in detailed recipe page.	1	Fix the logic to the following: as long as user has an ingredient in the recipe, it will return true instead of checking every ingredient user has.





