

# Testing

## Elton Lee

Tested the delete, preliminary add, and preliminary search functions.

- After hitting delete on a recipe, the recipe should be removed from the database. This is observable in real time in google firebase.
- Add function adds a recipe to the database. Renamed to save function once user authentication was finished.
- Search function should be able to search the database and return a recipe that meets the requirements.

## Qiutong Li

Tested the diet type search function.

- Hitting the diet type parameter should return the corresponding recipes within that category.
  - Equivalence classes: SearchPage.recipeSearch\_OnClicked, SearchPage.clearFeild, SearchPage.ContainsAny, FirebaseHelper.GetRecipe, RecipePage.nextRecipe\_OnClicked, RecipePage.GoURL
  - Test cases selected: Gluten Free. The output should be list of recipes within the category "Gluten Free". If multiple recipes available, next button would let to the following recipes. URL on each recipe page could direct to the right webpage. Home button on the recipe page could start a new search.

Tested the calories search function.

- Hitting search after inputting into the calories parameter should return a recipe with the appropriate amount of calories.
  - Equivalence classes: SearchPage.recipeSearch\_OnClicked, SearchPage.clearFeild, FirebaseHelper.GetRecipe, RecipePage.nextRecipe\_OnClicked, RecipePage.GoURL
  - Test cases selected: -500. Tested wrong format of input parameter and the correct error message printed.
  - Test cases selected: 0200. Tested the converting method that padding zeros to the input parameter does not affect the input reading. The output recipes are all satisfy the calorie setted condition. All recipes in the database that satisfies this condition were retrieved successfully, which in this case, 8 in total.

Tested simultaneous search functionality.

- Filling two and more search fields at a time, should all return recipes with corresponding search requirements.

- Equivalent classes: SearchPage.recipeSearch\_OnClicked, SearchPage.clearFeild, SearchPage.ContainsAny, FirebaseHelper.GetRecipe, RecipePage.nextRecipe\_OnClicked, RecipePage.GoURL, Checkup.ContainsAny
- Test case for two fields: Diet type: Vegan & Desired ingredient: tuna. Contracting error message printed. Class Checkup.ContainsAny worked.
- Test case for two fields: Desired ingredient: Chicken & Max calories: 500. All recipes contains chicken as one of the ingredients and also calories lower than or equal to 500 were outputted. (4 in total)
- Test case for three fields: Diet type: Gluten free & Desired ingredient: olive & Max calories: 600. All recipes that in category of gluten free, containing olive, and calories lower than or equal to 600 were outputted. (6 in total)
- Test case for four fields: Diet type: Vegetarian & Desired ingredient: butter & Unwished ingredient: nuts & Max calories: 400. All recipes that in category of vegetarian, containing butter, no nuts contained and calories lower than or equal to 400 were outputted. (2 in total)

#### Gardner Mein

Tested the user authentication (Login and Sign Up).

- Entering the correct email and password should log the user into the app.
- The user should have an account once the sign up button is pressed and all the requirements parameters are filled out.

Tested the save function.

- Hitting the save button should add the recipe to the database.

#### Zeyu Gao

Tested the desired ingredients search parameter.

- Hitting search after inputting into the desired ingredients parameter should return a recipe with the appropriate ingredients:
  - Equivalence class and function: SearchPage.recipeSearch\_OnClicked, SearchPage.clearFeild, SearchPage.ContainsAny, FirebaseHelper.GetRecipe, RecipePage.nextRecipe\_OnClicked, RecipePage.GoURL
  - Use "beef" as test case, the input will be first tested for its format, then will give us a list of recipes contain beef, use Click next to move to next valid recipe, and click the url to access the website.

Tested the unwished ingredients search parameter.

- Hitting search after inputting into the unwished ingredients parameter should return a recipe with the unwished ingredients:

- Equivalence class and function: SearchPage.recipeSearch\_OnClicked, SearchPage.clearFeild, SearchPage.ContainsAny, FirebaseHelper.GetRecipe, RecipePage.nextRecipe\_OnClicked, RecipePage.GoURL
- Use "chicken" as test case, the input will be first tested for its format, then will give us a list of recipes do not contain chicken, use Click next to move to next valid recipe, and click the url to access the website.

Michelle Yeung

Tested the next recipe button functionality.

- Pressing the next recipe button should take user to the next applicable recipe until there are no more recipes that fit the specifications
  - Equivalence classes:
- If no recipes are found, user should be notified that there are no more recipes in their specified parameters