Team 16 ¼ Cup Sprint 2 Week 1 Testing

Design Inspection

Product	¼ Cup Design Inspection
Date	3/4/18
Author	Team 16

Defect #	Description	Severity	How corrected
1	When creating the new page for recipes there was an error that we found by not reloading the list items page after calling the recipe.	2	We were able to fix this error by calling the loading of the page again after calling the recipe.
2	When creating the ability to view category sorts, we planned for separate view models for both the list of category sorts, and the list of categories inside them. These view models overlapped unnecessarily.	3	We removed the view model for the list of categories and kept the view model for the category sorts.
3	GUI for adding shopping list items was in the way of the items list that is populated bellow each item	3	The add item bar was moved up into the upper view to resolve this issue.
4	The keyboard popup is still persisting through fragment changes	3	Disable the keyboard popup when selecting a new fragment. (**In Progress)**
5	UI rendering and passing Recipe Information were being handled on the same thread.	2	RecipeViewModel was added to help “unblock” the UI thread on the RecipeFragment Module.
6	When pressing the back button from the recipe details fragment, the app exits rather than returning to the recipe fragment.	3	This was resolved by adding the recipe fragment to the back stack when performing the fragment transaction that swaps in the recipe details fragment.

Code Inspection

Product	¼ Cup Code Inspection
Date	3/4/18
Author	Team 16

Defect #	Description	Severity	How corrected
1	When trying to access the category sort list view model in the category sort fragment, it would create a new instance of the view model instead of getting the current one. This would cause issues when trying to access the sort order from the previous fragment.	2	Instead of obtaining the view model in the onCreate(), it was obtained in the onAttach() method of the fragment.
2	When trying to implement the category repository, there were many issues with trying to integrate it to the Dagger dependency injection, causing the application to not even compile.	1	The appropriate additions were made to the Dagger modules to enable injection for the category sort fragments and activity.
3	After creating the recipe search module, it was noticed that due to dependencies the app would crash whenever we tried to load it.	1	By adding the correct calls to the static AndroidInjection methods we were able to fix the issue.
4	Ingredient class couldn’t be accessed by the the RecepieInfoViewAdapter because it was not declared as a public class.	3	By declaring the Ingredient class as a public class in its own file, we were able the contents from the RecepieInfoViewAdapter.
5	The RecyclerView that displays the recipe ingredients was not displayed in the RecipeInfoFragment.	1	This was caused by generated code that was not updated when the fragment layout was changed, causing the recyclerView and its LayoutManager to not be instantiated. The generated if

			statement was removed in order to fix this issue.

Unit Testing

Automation:
We currently have not made automated tests for the new modules.

Modules from Sprint 1:

- Data Validation
- Shopping List Repository
- Shopping List View Model
- Shopping List UI
- Recipe List
- Welcome Message

Modules that our code relies on:

- Category Sort Module
    - Input: When a user applies a sort order to a list of items, the category sort module is used. For now, our sort algorithm takes a list of integers as the shopping list, and a list of sort categories from the sort order.
    - Output: The module takes both of these lists, and based off of the priority of the sort categories (the order of the categories in the sort order), will return a sorted shopping list.
- Search Recipe Module
    - Input: The user is able to search through the recipes in the Edamam API database. The user can use these recipes that the database returns to then add the ingredients to their shopping lists. (API call and adding to list in progress)
    - Output: This module will output the recipes from the API and then output the ingredients that the user wishes to add into their shopping list. (Tested using dummy values, not the API yet)

Product	¼ Cup Unit Testing for Category Sort
Date	3/4/18

Author	Team 16

Defect #	Description	Severity	How corrected
1	The initial implementation of the CategoryComparator sorted items with unknown categories to the beginning instead of the end of the list and sometimes sorted the same array in a different order.	3	The Comparator was modified to check if either argument was not a known category before comparing the indices of the categories, and sort it to the end of the array if necessary.
2	When adding a category to a category sort order after previously deleting one, the added category would be added with the red delete graphic instead of the text. The correct text shows up after holding down on the category, or when trying to move it.	3	(**In Progress)** This was fixed by resetting the RecyclerView adapter whenever a new item is added. The root cause is still under investigation.
3	The app crashed when launching the category sort settings on devices with API level lower than 23.	1	This was caused by the PreferenceFragment, which is not part of the support library, so the method onAttach(Activity), which was deprecated in API level 23 needed to be overridden in order to properly function on older versions of Android.

Product	¼ Cup Unit Testing Search Recipe
Date	3/4/18
Author	Team 16

Defect #	Description	Severity	How corrected
1	App crashed due to a Null pointer exception when trying to load in Stub recipe searches.	1	Initialized the Ingredients ArrayList in the Recipe object before for loop added text values to the names of the

			Ingredients.
2	The RecipeInfoFragment was not appearing after a recipe was being selected form the RecipeFragment ListView	1	The method to open the RecipeInfoFragment was being called from the RecipeFragment and not MainActivity. Added a call back function to fix this error.
3	Pressing the back button on the RecipeInfoFragment is closing the application.	3	Added the recipe fragment to the back stack when performing the fragment transaction that swaps in the recipe details fragment.
4	RecipeInfoFragment was not displaying a title value for the recipe that was selected.	3	The Id of the TextView for the title was name the same as another title TextView for a different fragment. Changed the Id to be unique.
5	Category sort orders are publicly visible and can be modified by all users.	2	(**In Progress)** Once category sort order creation is implemented, category sort orders should be queried by owner user ID and the requisite firebase security rules will be added to ensure that this pattern is followed.