Team MergeSort(); Project Milestone 5

By: Jack (John) O’Fallon, Henry Wang, Josh Meier, Gavin Zimmerman, Brandon Walker-Allen, & Alan Bourgeois

Title: HungryHill

- Swiping: We will upload random recipes to the database and make sure that the user can properly swipe through all of the recipes without issues. This will make sure the animations look proper on different sizes of screens, as well as make sure left swipes never pop back up for the user, right swipes are saved to the users favorited recipes page, and swipe up shows all the correct information about the recipe. We will be making sure the information on the swipe up is correctly correlated with each recipe. Specific cases include swiping right and making sure it gets stored in the correct screen. Another test includes a left swipe to see if that data always gets thrown out and won’t be shown again, it will clearly fail if we see the same recipe again within our test data set. These tests will prove the correct functionality based on input and output, as well as making sure the app can handle the function without crashing.

- Filters: We have a predetermined data set that we can insert into the code, of which we know what the outcomes should be, and compare that with what actually is being output. As for the specific filters we will have different users select different criteria for their specific diets and make sure only the correct recipes show up for them. We will use log.console to verify our background processes. We will also make sure to include edge cases where all or none of the filters are selected. A specific case includes the functionality of our hamburger menu drop down, and that the filters chosen are actually applied for the user’s recipes. These tests will prove the correct functionality based on input and output, as well as making sure the app can handle the function without crashing.

- Ranking: We have a predetermined data set that we can insert into the code, of which we know what the outcomes should be, and compare that with what actually is being output. As in rankings, we will have scores from 0 – 5 for each recipe and make sure that they are displayed properly when the app is run. The user will swipe through this pre-set of data and make sure all data matches up properly. We will use log.console to verify our background processes. These tests will prove the correct functionality based on input and output, as well as making sure the app can handle the function without crashing.

- We are looking into pushing a beta test with 20 – 50 users to make sure our servers can hold a medium sized capacity without crashing. Also, this will help us find other bugs in our code as we will be asking all individuals involved for any further feedback.