

System and Unit Test Report

Product Name: Foodie More

Team Name: Four Gangs

Date: 6/5/17

Sprint 1 Test:

User Story:

As a user, I am able to register a user account via the Android app, and view my personal account.

Scenario:

1. Start Foodie More App; Select "Sign Up" Button; Select CONSUMER Button, type:
 - Email = (EditText) username;
 - Password = (EditText) password;
 - confirmPassword = (EditText) confirmPassword
2. Select Done Button
User should be able to see the layout that indicates he has successfully signed up
3. Touch the Screen
User will be led to the main page. The user Info is located in the action Bar Menu
4. Touch the Action Bar Menu, Select Setting Option
User will see his personal information and be able to update his info

User story:

As a seller, I am able to register a seller account via the Android app, and view my personal account.

Scenario:

1. Start Foodie More App; Select "Sign Up" Button; Select MERCHANT Button, type:
 - Email = (EditText) username;
 - Password = (EditText) password;
 - confirmPassword = (EditText) confirmPassword
2. Select "Next" Button
3. User can see the next part of signing up, type:
 - Restaurant Name = String resName
 - Restaurant Street address = String restStreet
 - Restaurant phone number = String resPhone
4. Select "Done" Button
5. User should be able to see the layout that indicates he has successfully signed up
6. On the top of screen, there are four buttons. Select the second one which is personal information button.
7. User will see his personal information and be able to update his info

User story:

As a user, I am able to view the product listed by the sellers.

Scenario:

1. Start Foodie More App; type:
UserID = (EditText) username
Password = (EditText) password
2. Touch "LOGIN" button
3. User will be able to see all restaurants which includes the name, rating, and current waiting time.
4. Touch any restaurants, user will be led to detail of selected restaurant. In this page, other than information provided from last page; user will see restaurant address through Google Map; all dishes with different categories and corresponding prices.

User story:

As a back-end developer, I need to build a user database to store all the user information and Dishes information.

Scenario:

1. In the firebase database, the data is divided into three parts:
Accounts, which store all the UID of the registered users or merchants and each uid has either merchant or user value:
Every time user and merchant register a new account, the sign up information will be stored in Firebase Authentication and User UID will be stored under account root.
RestaurantData, which is divided further by different restaurants, and in the each restaurant, there would be MenuInfo, Order, and ResInfo options
In merchant page, a merchant can update his menuInfo and Restaurant information to firebase database under this uid.
UserData: contains the user Information
After sign up with Firebase Authentication, user information will be stored under UserData.
2. We have no protocol used by firebase database

Story Point Not working:**User Story:**

As a back-end developer, I need to create APIs on the server side for the front-end app.

Situation:

After group discussion, we abandon the server side, and simply using firebase to implement all of our functions.

Sprint 2 Test:

User story:

As a back-end developer, I need develop the searching algorithm and searching option.

Scenario:

1. In the restaurant list page. Click “magnify lens” picture on the top right.
2. User will see the textview to enter the key words. After entering the requirement, user will see the restaurants that meets the requirements.
3. Click cancel, it will go back to the main page.
4. Click “funnel” shape picture
5. User will see the two sorting ways, one is based on waiting time, another is based on the rating.
6. Click waiting time option, the restaurants will be listed by waiting time from low to high; Click rating option, the restaurants will be listed by rating from high to low.

User story:

As a user, I am able to add the dishes into my shopping cart.

Scenario:

1. In the detail of restaurant page of Dominos Pizza page. user can choose breakfast, lunch and main
2. User choose “main” option
3. User should see all the food that has attribute “main” course. On the right of one dish, there is a button having the price as label. Click “pizza” button.
4. One pizza will be added in the cart.
5. Click “pizza” button
6. One more order of pizza will be added in the cart
7. Click “shopping car” picture on the upper right corner
8. It will go to the check-out page. In the check-out page, user will see pizza order with quantity = 2 and unit price is \$5

Story Point Not Working:

As a front-end developer, I need to use APIs provided by the server to send requests and receive responds. Then I make the front-end app update UI and internal data structures according to the returned data from the server.

Situation:

After group discussion, we abandon the server side, and simply using firebase to implement all of our functions.

Sprint 3 Test:

User story:

As a developer, I need to add current waiting time information so that the seller can update and user can view while they are browsing the App.

Scenario:

1. Log in a consumer account.
2. Selected Domino restaurant. The current waiting time is in upper-left part of screen
3. Touch the back button
4. Touch the Action Bar Menu, select “Sign out” option
5. Log in the Domino account, user will see the main restaurant page.
6. On the top of screen, there are four buttons. Select the third one which is notification button.
7. Type a number and touch “UPDATING WAITING TIME” button
8. Select the setting button which in the right of notification button.
9. Touch the Action Bar Menu, select “Log out” option.
10. Log in a consumer account again and select the Domino restaurant.
11. User will see that the current waiting time has change to the number typed in step 7.

User story:

As a user, I am able to check out the order.

Scenario:

1. In consumer main page, select the Domino restaurants.
2. Touch price button of Pizza
3. Touch the shopping car button which is in the upper-left part of screen.
4. User will see selected dish “Pizza”, its quantity, and its total prices.
5. Touch the “pay” button, user will go to payment page.
6. Touch the “Pay with PayPal” button whose color is blue
7. Type:
Email = “foodiemore@pay.com”
Password = “12345678”
8. Touch “Login” button
9. User will see the total price of all dishes
10. Touch “Pay” button. The payment has been completed and the page will return to Consumer main page.

User story:

As a seller, I am able to accept the order and send an order information

Scenario:

1. Log in a consumer account, select Domino restaurant, and check out an order with some Pizza.
2. Log in a merchant account and select notification button
3. User will see that there exists an order checked out in step 1.
4. Touch the “VIEW” button and user will see the detail of order including dish name “Pizza” and its quality.
5. Touch “DELETE” button to remove this order.

User story:

As a developer, I am able to use the Google API to add distance to current location.

Scenario:

1. Log in a consumer account, select Domino restaurant
2. User will see that there is Google map in the center of screen
3. In the map, the red line represents a path from current location to Domino.