Test Cases and Requirements Test Coverage for Foodify

Anandarajan Sindini - Project Manager

Kolady Anamika Martin - Release Engineer and QA Manager

Duong Ngoc Yen - QA Engineer

Verma Nandini - Lead developer and Front-end developer

Chua Jim Sean - Backend developer

Team RunTime Terror

School of Computer Science and Engineering

Nanyang Technological University, Singapore

Version History

Version	Implemented By	Revision Date	Approved By	Approval Date
1.0	Duong Ngoc Yen Anandarajan Sindini Chua Jim Sean	27/10/2022	Anandarajan Sindini	30/10/2022

Contents

Purpose of Document

Test cases are a set of actions that are planned to verify the execution of particular features or functionalities of a software application. The test cases that will be undermentioned are specially designed to test the various functionalities of Foodify, a web application which aims to assist its users in mitigating their food wastages. These test cases will determine whether the Foodify system performs in a way that it satisfies all standards, guidelines and initial requirements.

Test case writing is a very crucial part of software testing. The process of writing test cases can also reveal errors or defects within the software. Hence, it is utilised by various teams such as the testing team, development team and management team. Test cases connect the whole testing process together. Properly developed test cases can best measure the customer satisfaction of a software system. It is the most accurate way of checking whether whatever specified requirements are fulfilled. While implementing the test cases, we may come across more defects that might have been skipped in ad-hoc testing.

Adding on to test cases, test coverage is a metric in software testing that is used to measure the amount of tests performed by a set of test cases. This will include gathering information about the program that is being executed when running the test suite to find out which branches of conditional statements have been taken. Requirements test coverage is used to specifically cover how well the software requirements are covered by the test cases. We covered all requirements by test configurations instead of covering each requirement only at the level of the test. A test configuration represents a specific use-case of a test.

In this document, all our test cases which were formed by the testing team are listed. These test cases will verify that Foodify fulfils our product requirements and standards. The tests are divided into 3 main categories: unit tests, integration tests and system tests. The unit tests will be conducted first, followed by the integration tests and, finally, the system tests.

Unit Testing

Test Case 1

Test Case #	UT1	**Test Name**	Side Bar
System	Foodify	**Subsystem**	N/A
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	There are 6 pages in the side bars: Home, Add food items, my food items, sell food items, explore recipes, marketplace. When the user clicks on a tab, the system should navigate the user to the respective pages.		

Pre-conditions	1. The user is logged in

Step #	**Action**	**Expected System Response**
1	Tap on Home tab on the side bar	The system navigate user to Home page
2	Tap on Add food items tab on the side bar	The system navigate user to Add food items page

3	Tap on My food items tab on the side bar	The system navigate user to My food items page
4	Tap on Sell food items tab on the side bar	The system navigate user to Sell food items page
5	Tap on Explore recipes tab on the side bar	The system navigate user to Explore recipes page
6	Tap on Marketplace tab on the side bar	The system navigate user to Marketplace page

Post-conditions	The user is in the respective pages.

Test Case 2

Test Case #	UT2	**Test Name**	Page Navigation
System	Foodify	**Subsystem**	Home Page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	There are 6 pages accessible from the homepage: My Profile, View my Items, Visit Marketplace. When the user clicks on a tab, the system should navigate the user to the respective pages.		

Pre-conditions	1. The user is logged in 2. User is in Home page

Step #	**Action**	**Expected System Response**
1	Tap on My profile button	The system navigate user to Profile page
2	Tap on View my items button	The system navigate user to My Food Items page
3	Tap on Visit marketplace button	The system navigate user to

	the Side Bar	Marketplace page

Post-conditions	The user is in the respective pages.

Test Case 3

Test Case #	UT3	**Test Name**	Logout Button
System	Foodify	**Subsystem**	Home Page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	When user click on the log out button, the system must navigate user to the start page, and log out from the system		

Pre-conditions	1. The user is logged in 2. User is in Home page

Step #	**Action**	**Expected System Response**
1	Tap on Logout button	The system navigates the user to the Start page and logout from the system.

Post-conditions	The user is logged out

Test Case 4

Test Case #	UT4	**Test Name**	Choose food items
System	Foodify	**Subsystem**	Post Items
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	When user click on the Food items input in selling post page, there is a drop down for the user to choose. The drop downs must contains all the food items the user is currently having		

Pre-conditions	1. The user is logged in. 2. The user is in sell food items page.

Step #	**Action**	**Expected System Response**
1	Tap on the choose food item drop down	The system displays a list of all items the user is currently having

Post-conditions	The user could choose an item from the user’s food items list

Test Case 5

Test Case #	UT5	**Test Name**	Chat button
System	Foodify	**Subsystem**	Marketplace page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	When user click on the chat button under a post, the system directs the user to the chat page, with the respective information from the post and the seller information displayed.		

Pre-conditions	1. The user is logged in. 2. The user is in the marketplace page

Step #	**Action**	**Expected System Response**
1	Tap on the chat button under a post	The system direct the user to the chat page, with the respective information from the post and the seller information displayed.

Post-conditions	The user is in the chat page, and with information from the item

	and the seller.

Integration testing

An incremental top-down approach has been used to conduct integration tests on Foodify. This ensures that usability is not compromised for edge cases and when the use case involves a long sequence of actions.

Test Case 1

Test Case #	IT1	**Test Name**	Sign Up
System	Foodify	**Subsystem**	Start page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user must be able to register a new account using Firebase Authentication		

Pre-conditions	1. User must register with an unregistered email address.

Step #	**Action**	**Expected System Response**

1	Tap on the Sign up button	User must see the sign up page with respective inputs fields for the user to fill in.
2	Enter email, password, name, phone number	The system records and displays the input information in the input fields
3	Click on Sign up	The system register user as a new account and navigate to the home page with the new account.

Post-conditions	The user is logged in and in the home page.

Test Case 2

Test Case #	IT2	**Test Name**	Login
System	Foodify	**Subsystem**	Start page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user logs in to the system using existing account		

Pre-conditions	1. The user must have an existing account

Step #	Action	Expected System Response
1	Tap on the Login button	User must see the log in page with respective inputs fields for the user to fill in.
2	Enter email, password.	The system records and displays the input information in the input fields
3	Click on Log in button	The system verify if the user is existing and navigating the user to home page.

Post-conditions	The user is logged in and in the home page.

Test Case 3

Test Case #	IT3	**Test Name**	Logout
System	Foodify	**Subsystem**	Start page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user is logged out from the system.		

Pre-conditions	1. The user is logged in

Step #	**Action**	**Expected System Response**
1	Tap on the Logout button	The system log out user from the current runtime and navigate to the Start page

Post-conditions	The user is logged out and in the start page.

Test Case 4

Test Case #	IT4	**Test Name**	Change Password
System	Foodify	**Subsystem**	Profile page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	User change password from old to new password and is able to log in using the new password		

Pre-conditions	1. The user is logged in 2. The user is in profile page

Step #	Action	Expected System Response
1	Tap on the Chang password button	The system navigate user to the change password page.
2	Enter old password and new password	The system displays the old password and new password in the input fields.
3	Click on Change password button	The system update the user's password

Post-conditions	User's password is updated.

Test Case 5

Test Case #	IT5	**Test Name**	Edit Account
System	Foodify	**Subsystem**	Start page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user could change the profile, including: Name, Phone number		

Pre-conditions	1. The user is logged in 2. The user is in Profile page

Step #	**Action**	**Expected System Response**
1	Tap on the Edit account button	Sytem navigate user to the edit account page
2	Enter new name, phone number	The system records and displays the input in the input fields.
3	Click on Update account	The system update user’s information in the database

Post-conditions	User’s information is updated in the database

Test Case 6

Test Case #	IT6	**Test Name**	Add Food Item
System	Foodify	**Subsystem**	My food item
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022

Description	The user is able to add new food items to the current food items list

Pre-conditions	1. The user is logged in

Step #	**Action**	**Expected System Response**
1	Click on Add food item button	Sytem navigate user to the add food item
2	Enter Food item information: name, quantity, expiry date	The system records and display the input the the input fields
3	Click on Add food item button	New food item is added to the database with the user id is the current user id.

Post-conditions	New food item is added to database

Test Case 7

Test Case #	IT7	**Test Name**	View Food Items
System	Foodify	**Subsystem**	My food item
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user views all the current food items with respective expiry dates		

Pre-conditions	1. The user is logged in

Step #	**Action**	**Expected System Response**
1	Click on My items button	The system navigates user to my food items page and renders all food items of the current user with respective expiry date and displays on the screen

Post-conditions	All food items are displayed on the screen.

Test Case 8

Test Case #	IT8	**Test Name**	Post Food Items
System	Foodify	**Subsystem**	Marketplace page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user post a food item into marketplace		

Pre-conditions	1. The user is logged in 2. The user is in the marketplace page 3. The user has at least one food item in the My food Items list

Step #	**Action**	**Expected System Response**
1	Click on Post on marketplace	The system navigates user to the sell food item page
2	Enter Food item post information: title, description, food item, status, price	The system records and display the input the the input fields
3	Click on Post button	New food item post is added to the

		database.

Post-conditions	New food item post is added to database

Test Case 9

Test Case #	IT9	**Test Name**	View Marketplace
System	Foodify	**Subsystem**	Marketplace page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user views all post from the marketplace		

Pre-conditions	1. The user is logged in 2. The user is in the marketplace page

Step #	**Action**	**Expected System Response**
1	Click on visit marketplace button	The system navigates the user to the marketplace page and renders all food items posted by all users.

Post-conditions	New food item is added to database

Test Case 10

Test Case #	IT10	**Test Name**	Contact Seller
System	Foodify	**Subsystem**	Marketplace page
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user is interested in a food item and want to contact the food seller.		

Pre-conditions	1. User is logged in 2. User is in marketplace page

Step #	**Action**	**Expected System Response**
1	Click on chat icon at the end of the post	System navigates user to the chat page, displays post information (Item name, expiry date, price, status, description) and seller information (name, email, phone number) and

		payment button.
2	Click on the Payment button	System displays the successful status of the payment, item of the seller is deleted from the seller food items list.
3	Click on the cancel button	System navigate user back to marketplace page

Post-conditions	User could view and/or buy the food item.

Test Case 11

Test Case #	IT11	**Test Name**	Recipes Recommendation
System	Foodify	**Subsystem**	My food item
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The system recommends 5 recipes from the current food items of the users.		

Pre-conditions	1. User is logged in

	2. User is in my food item page

Step #	**Action**	**Expected System Response**
1	Click on Recipes recommendation button	The system navigates the user to the Recipes recommendation page, using spoonacular api to get the recipes, with the input ingredients are the ingredients from user food items, displaying all ingredients and image of each recipe.

Post-conditions	All recipes are rendered and displayed correctly in recipes page

System testing

System testing is the process of testing all the integrated hardware and software components of our system to verify that the application meets its specified requirements.

Test Case 1

Test Case #	ST1	**Test Name**	Smoke testing
System	Foodify	**Subsystem**	Smoke testing
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The user must be able to log in and navigate across and use all features designed		

Pre-conditions	1. Use has a valid account

Step #	**Action**	**Expected System Response**
1	View profile	System navigates user to the Profile page and displays user information.

2	Edit profile	System navigate user to edit profile page and update profile with the input information
3	Change password	System navigate user to change password page and update user’s password with the new password.
1	View food item	System navigates user to the view food items page and displays all food.
2	Add food item	System navigates user to add food item page and adds the input information as new food item to the database
3	View recommended recipes	System navigates user to the recipes recommendation page and displays recommended recipes, using user’s food items as input ingredients.
4	View marketplace	System navigates user to the marketplace pages and extract posts from the database and displays all posts.

5	Add marketplace post	System navigates user to the sell food item post, create a new post with user input and add to the database.
6	Contact seller	System display food item that user are interested, seller contact information and display payment status, delete seller's sold item

Post-conditions	The complete system is able to interact based on user's action.

Test Case 2

Test Case #	ST2	**Test Name**	Stress testing
System	Foodify	**Subsystem**	Stress testing
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	Multiple user must be able to use the application simultaneously		

Pre-conditions	1. All users must have a valid Foodify account.

Step #	Action	Expected System Response
1	20 users is using Foodify simultaneously	The application should work smoothly without crashing. The users should be able to navigate across pages and be able to use all the features of Foodify.

Post-conditions	The system is operated by 20 users simultaneously.

Test Case 3

Test Case #	ST3	**Test Name**	Scalability testing using Database
System	Foodify	**Subsystem**	Scalability testing using Database
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	Database should be able to process queries from multiple application without crashing		

Pre-conditions	1. All users must have valid Foodify accounts.

Step #	**Action**	**Expected System Response**
1	20 users is using Foodify simultaneously. The users must edit and use the database required function	The database is updated simultaneously in realtime. The update in database should be smooth and the server should not crash.

Post-conditions	The database is updated based on the application usage of 20 users simultaneously.

Test Case 4

Test Case #	ST4	**Test Name**	Recoverability Testing
System	Foodify	**Subsystem**	Recoverability Testing
Designed by	QA Team	**Design date**	30/10/2022
Executed by	QA Engineer	**Executed date**	30/10/2022
Description	The data stored for any user must be stored in the database if		

	the app crashes or the device loses internet connectivity.

Pre-conditions	1. The device must be connected to stable internet connection.

Step #	**Action**	**Expected System Response**
1	The user is viewing the food items and the system disconnects to the internet	The database must contain all user’s confirmed actions. All added food items, posts, profile information are stored and updated in the database.

Post-conditions	The application is able to recover the data of the user in case the app crashes while its operation.

Summary

Test case	Test data	Expected result	Actual result	Status
UT1	NA (UI Flow Testing)	The system could navigate user to respective pages	The system navigates user to respective pages	Pass
UT2	NA (UI Flow Testing)	The system could navigate user to respective pages from home page	The system navigates user to respective pages from home page	Pass
UT3	NA (UI Flow Testing)	The system log out user from the system	The system log out user from the system	Pass
UT4	Choose the first food item from the list	The system should show the respective food item	The system shows the respective food item	Pass
UT5	NA (UI Flow Testing)	The system should navigate user to contact page with seller	The system navigates user to contact page with seller	Pass
IT1	Email: testinguser1 Password: 12345678 Name: testinguser1 Phone number: 123456789	The user could register a new account successfully	The user registers a new account successfully	Pass
IT2	Email: testinguser1 Password: 12345678	The user could log in successfully	The user logs in successfully	Pass
IT3	NA (UI Flow Testing)	The user could log out from the system	The user logs out from the system	Pass
IT4	Old password:	The user could	The user changes	Pass

	12345678 New password: 123123123	change password successfully and could log in using the new password	password successfully and could log in using the new password	
IT5	Name: user1 Phone number: 98765678	The user could update profile successfully	The user's profile is updated successfully	Pass
IT6	Name: coffee Quantity: 3 packs Expiry date: 04/11/2022	The user could add new items to the database	The new item is added to database	Pass
IT7	NA	The user could view all food items	All food items are retrieved and displayed	Pass
IT8	Title: Coffee to clear Description: Bought 3 months ago, could not finish. Food Item: 3 packs coffee, expiry 04/11/2022 Status: opened Price: 10	New post is added to the posts database collection	New post is added to the posts database collection	Pass
IT9	NA	All posts are retrieved and viewed in user's marketplace page	All posts are retrieved and viewed in user's marketplace page	Pass
IT10	Click on a post	Display post information, seller information, and payment status	Display post information, seller information, and payment status	Pass
IT11	NA	Display all recipes from the user's food items	Display all recipes from the user's food items	Pass
ST1	Smoke test	The user must be	The user must be	Pass

		able to log in and navigate across and use all features designed	able to log in and navigate across and use all features designed	
ST2	Stress test	Multiple users must be able to use the application simultaneously	Multiple users are able to use the application simultaneously	Pass
ST3	Scalability test using Database	Database should be able to process queries from multiple application without crashing	Database are able to process queries from multiple application without crashing	Pass
ST4	Recoverability Test	The data stored for any user must be stored in the database if the app crashes or the device loses internet connectivity.	The data stored for any user are stored in the database if the app crashes or the device loses internet connectivity.	Pass