QA Fundamentals and Manual Testing Exam Prep II

Part I - Foody Web Application

You are provided with a web application, **Foody**, along with a specification of the requirements. The primary objective of the exam is to ensure the App functions as expected according to the provided use cases. This involves **creating test cases for each use case**. It also includes the **detection and detailed documentation of any identified bugs**. You have to document all your work in the provided **Test Management and Bug Tracker Template**.

Software Requirements

1. Introduction

1.1. Purpose

The objective of this document is to provide a description of the Foody application (also referred to as Foody or The App). It will present an overview of the key functionalities.

1.2. Scope

This document includes high-level descriptions of the **basic functionalities of Foody**, such as user registration, login, profile editing, food creation and management. It does not cover any special user (Administrator) functionalities.

2. Overall Description

2.1. System environment

Foody hosts two primary actors: **unregistered/non-logged** users and **registered/logged users**. Both can access their respective parts of the application via the internet on the following url:

http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com:85

2.2. Key Features

Unregistered/non-logged users have access only to the Home page of The App, with the option to either **SIGN UP** or **LOG IN**. On the other hand, **registered/logged users** have access to the **main functionalities**, including **Profile editing**, **Food creation**, and **Food management**.

2.2.1. Home Page for unregistered / non-logged users

The **Home Page** is the main gateway into Foody App. It contains a **navigation bar** (also refered to as **Navbar**) **on top**, with **Foody Home Page** link on the left and **SIGN UP** and **LOG IN buttons** on the right.
The **Home Page is scrollable** and the central focus is to encourage users to share their food experiences. Below the main title, there is a **LEARN MORE** button, which leads to **About Us Page**.

Further below, there are four sections promoting The App experience. Each consists of motivational phrase, short description and corresponding picture: **Share your food**, **Describe Your Dish**, **Your Hub for Food Adventures**, **Sign Up Now**. The **picture** for **Sign Up Now** section, reads **Join US** and is clickable, leading to the **SIGN UP page**.
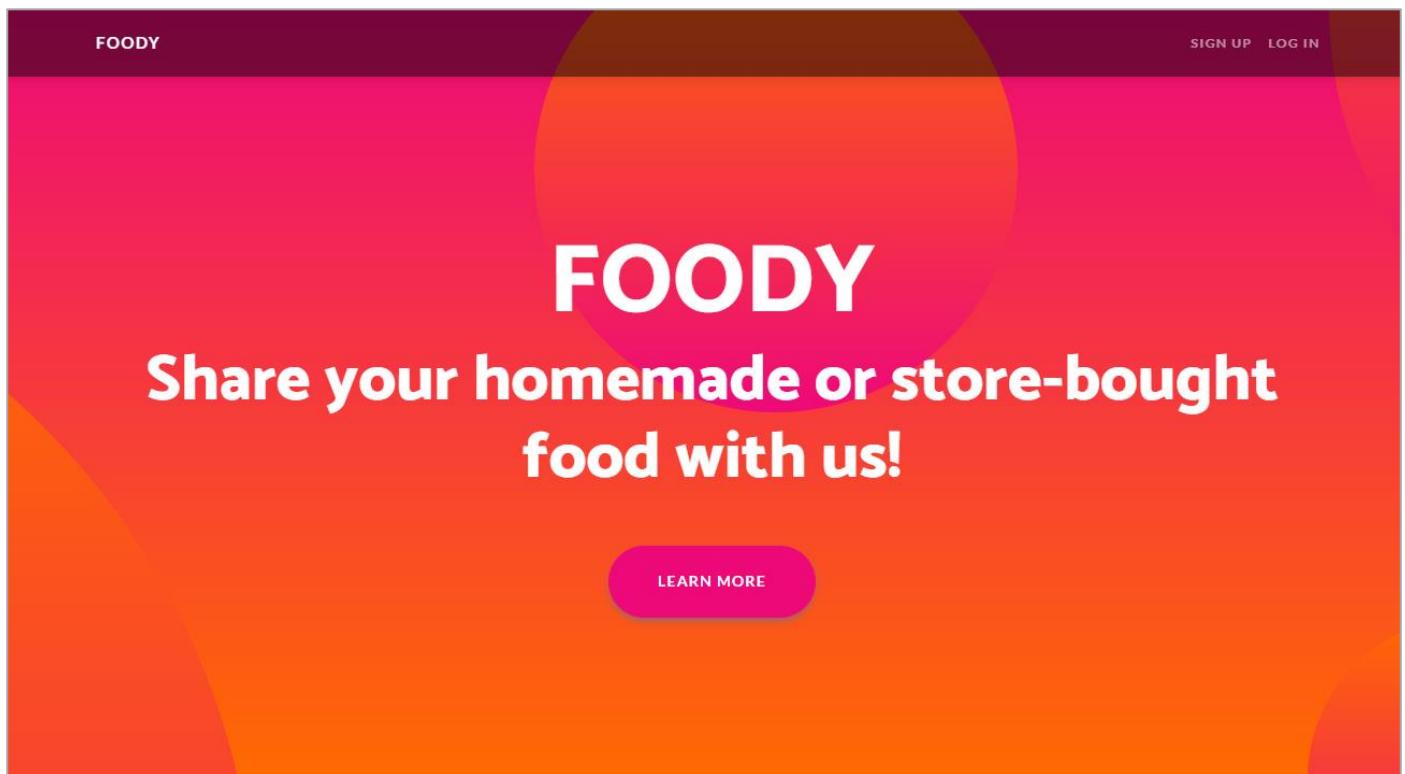
2.2.2. Sign Up Page

The **Sign Up page** is accessed from the Home page **via SIGN UP button** on the Navbar and from the **Join US picture**. It contains **registration form** with fields for entering **Username**, **Email**, **First Name**, **Middle Name**, **Last Name**, **Password**, and **Repeat Password**. The form also includes a hyperlink to view the Terms and Conditions document and a **SIGN UP button** to submit the form. There is also a **LOG IN HERE button** for registered users, which redirects to **LOG IN page**.

The acceptable requirements for each field on the registration form are specified as follows:

Username: A 2-30 character field, accepting all character types.
Email: Requires a valid email, 6-254 characters long.
First Name: A 2-60 character field, accepting all character types.
Middle Name: A 2-60 character field, accepting all character types. Not required.
Last Name: A 2-60 character field, accepting all character types.
Password: Any characters are acceptable, with a length of 6-30 characters
Repeat Password: Must match password.

2.2.3. Log In Page

The **Log In page** can be reached **from the Navbar LOG IN button on the Home page**. The page provides fields for **Username** and **Password**, as well as a **Forgot password** option, also a **LOG IN button** and a **CREATE NEW button**, which leads to **Sign Up page**.

Forgot Password functionality leads to a Restore password page, where the password can be restored by verified email.

2.2.4. Home Page for logged users

Once a user is **logged in**, they are directed to a different version of the **Home page**. The page includes a **Navbar on top**, with links to the **User's profile** and **Home page** on the left, and **ADD FOOD** and **LOGOUT buttons** on the right. The Home page includes **welcome message** with the **User's name**, and a **Search Field with a Search button**. If there is no food added, the **Home page** is showing **There are no foods :(** message and a **ADD FOOD button**, which leads to **Add Food page**.
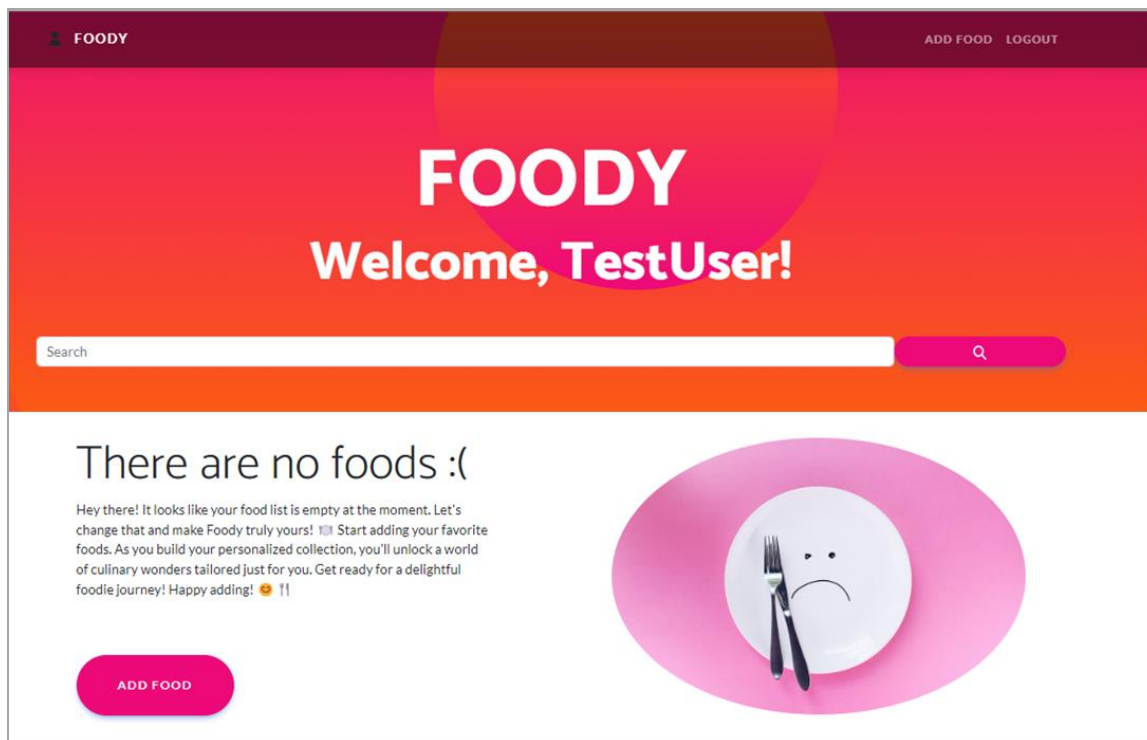
a. Home Page Search Functionality

The **search functionality** is available on **Home Page** and contains a **Search field and a Search button**. Users can enter **title** of the foods and click the Search button to initiate the search. The search functionality then filters the list of foods based on the provided titles. If there are no foods found, **There are no foods :(** message is shown again, as well as the **ADD FOOD button**, which leads to **Add Food page**.

2.2.5. My Profile Page

This page is accessible for logged users **from the Navbar, top-left** 👤 **icon**. It includes **four sections**. On the left, by default an **empty profile picture, Username and an EDIT button**. On the right are **User's requisites** - Full Name, Email and Total food revues counter (empty by default). Below User's requisites, are located the **About me section** (empty by default)and **About my food passion section** (also empty by default).
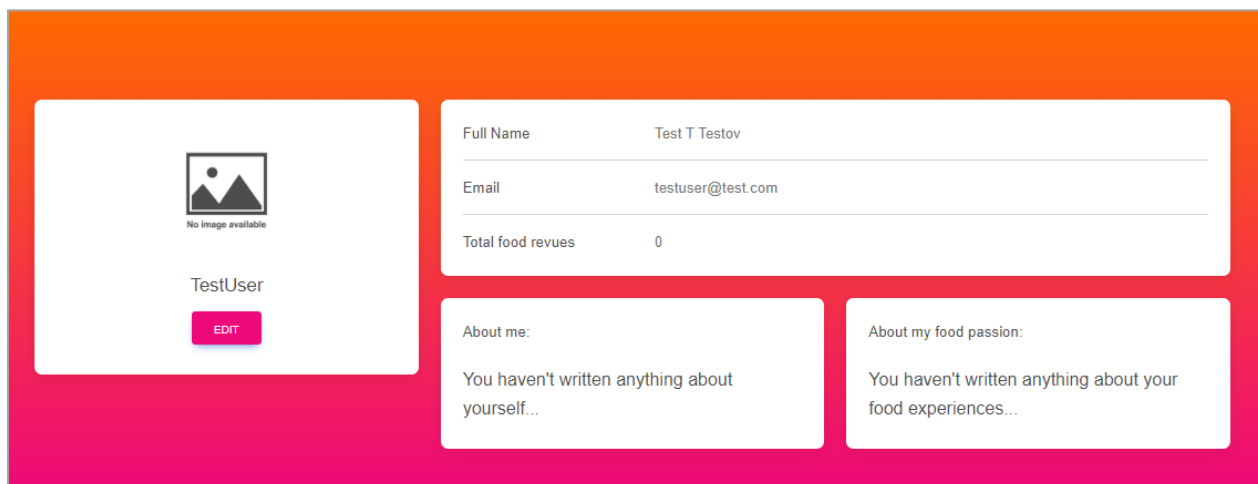
2.2.6. Edit Profile Info Page

Accessible from **EDIT button on My Profile page**, this page allows users to **edit their profile information**, including Profile picture (picture is not uploaded, but must be a valid URL of a picture), First name (max 60 characters length) , Middle Name (max. 60 characters length), Last name (max. 60 characters length), About me (max. 256 characters length), and a About my food passion section (max. 256 characters length).

2.2.7. Add Food Page

This page is accessible from the **Add Food link on the navbar** and contains the following: **Food name** field (max. 70 characters length), **Describe your food** field (max. 400 characters length), **Add a picture of your food** field (picture is not uploaded, but must be a valid URL of a picture) and **ADD button**.

Upon **creating a Food**, it redirects to **Home page for logged users**, where the new **added food is present** with it's attributes and buttons for **VIEW, EDIT, and DELETE**
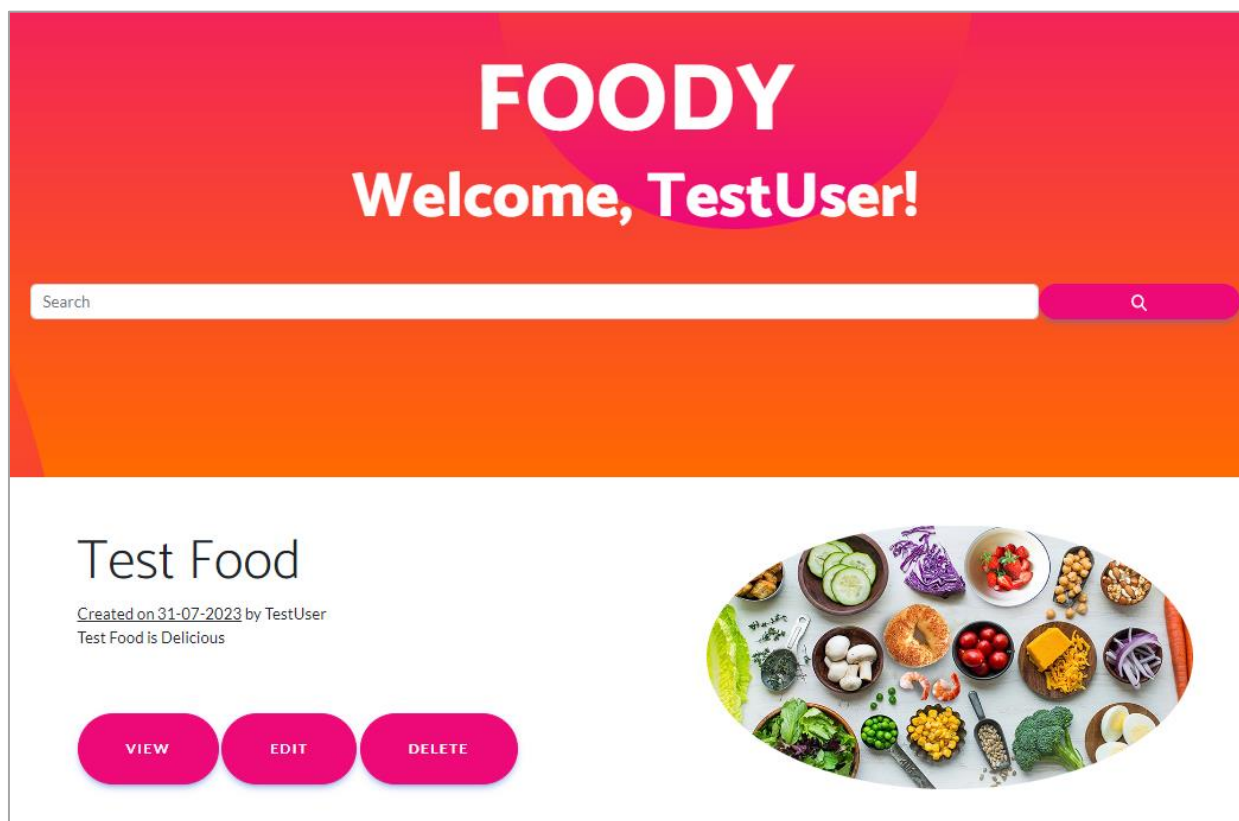
2.2.8. Food Management

Food management Functionalities are **available from the Home Page**. Each created Food has its own **VIEW, EDIT and DELETE buttons**.
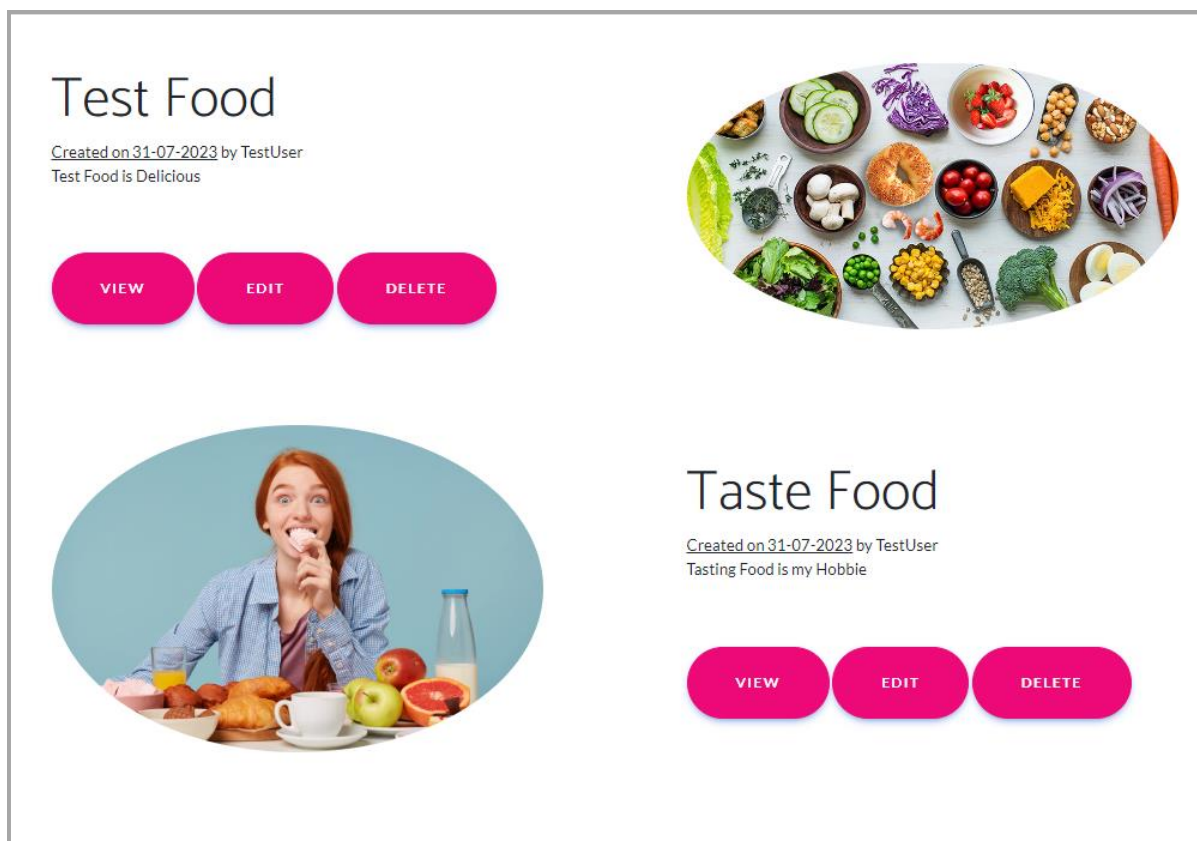
a. View

Available after clicking on VIEW button.
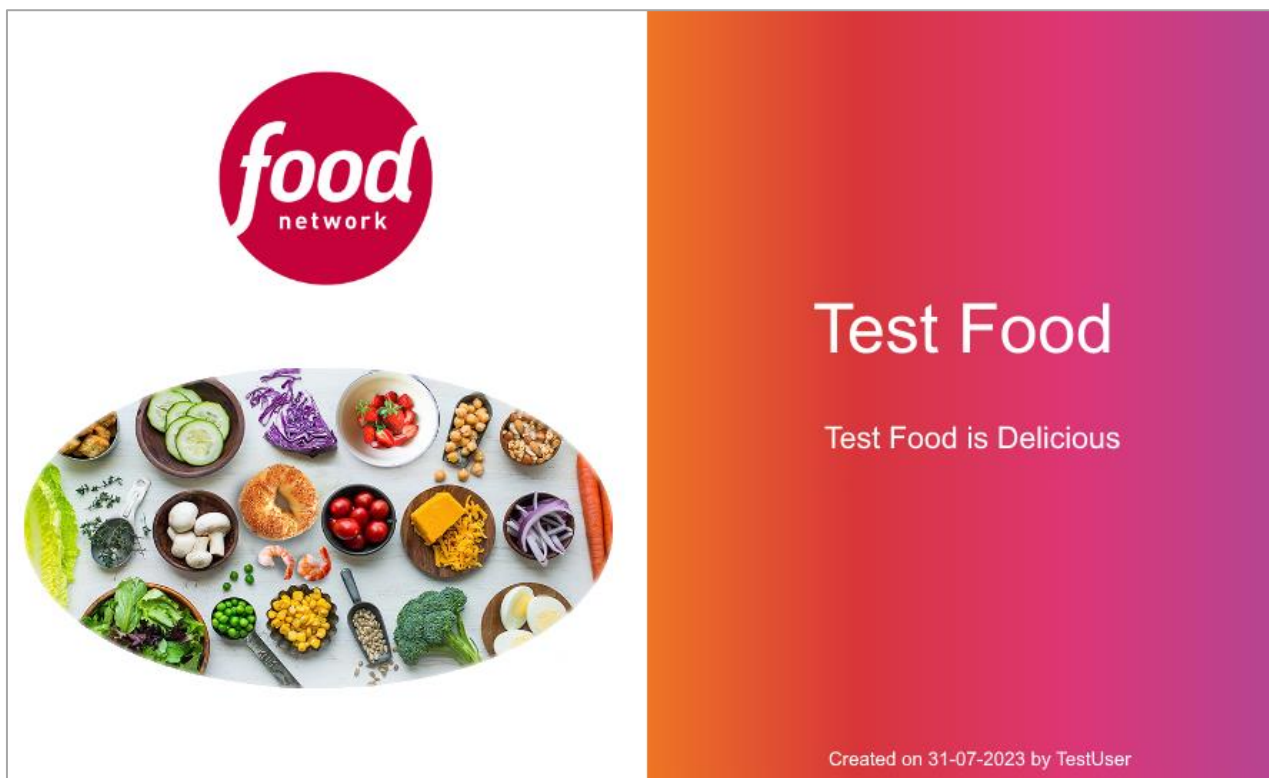
b. Edit

Available after clicking on **EDIT button**.
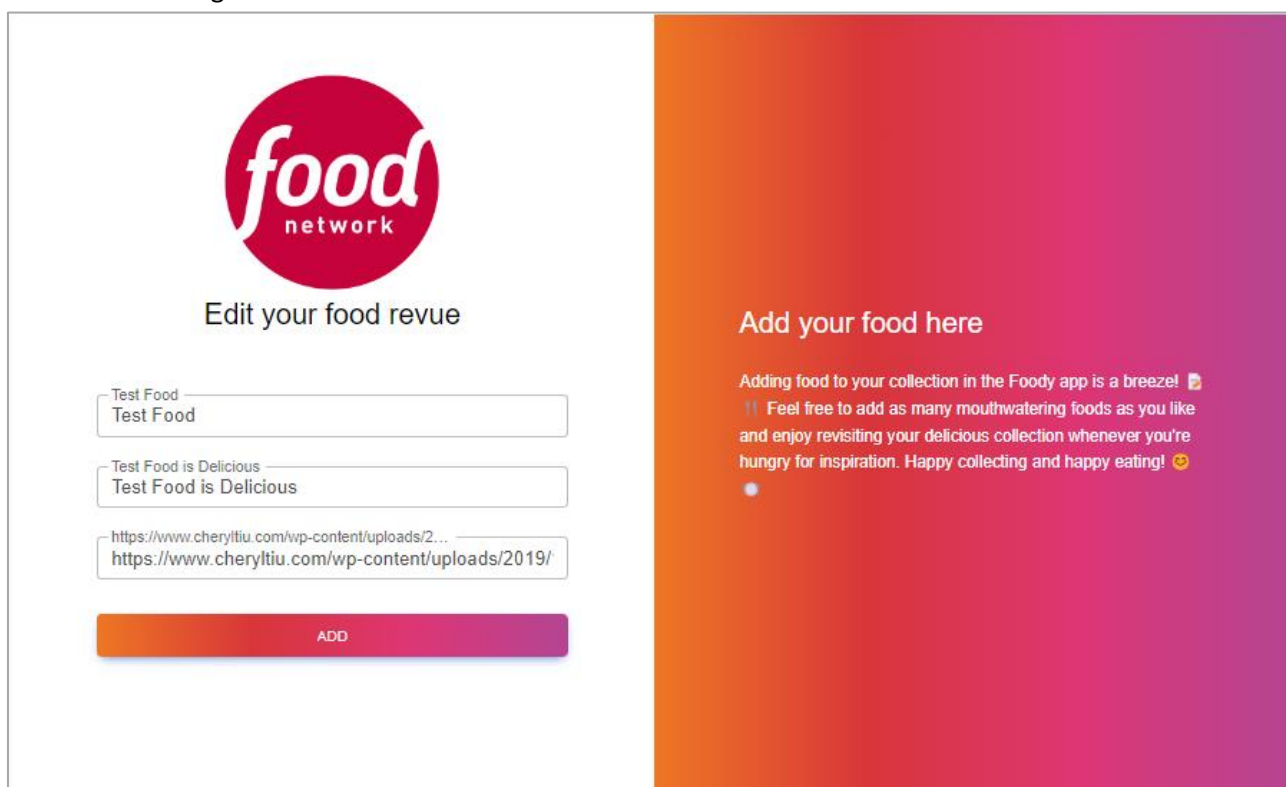
c. Delete

After clicking on **DELETE button**, the Food is deleted and it disappers from the Home page.

Functional Requirements

1. Use Case 1 (Home Page)

Users should be able to access The Foody App from its designated URL via the internet, which should load the Home page, appropriate to the user's logged-in status (unregistered/non-logged or registered/logged).

2. Use Case 2 (User Registration)

Unregistered users should be able to successfully go through the Sign Up process. This involves the utilization of fields such as Username, Email, First name, Middle Name, Last name, Password and Repaeat Password, all subject to their respective constraints and character type acceptance. User should have the option to read the Terms and Conditions and agree with it.

3. Use Case 3 (User Sign In)

Registered users should be able to successfully go through the Log In process. They should be able to log in using Username and Password.

4. Use Case 4 (Profile Management)

Upon successful Log In, logged users should be able to navigate to their Profile page, edit their profile details, and view their Foods count. This involves the changing of profile picture inserted via URL, and editing of user data fields including First Name, Middle Name, Last Name, About, and About Food.

5. Use Case 5 (Food Creation)

Logged users should be able to navigate to the Add Food page, where they can create a new food with a Title, Description, and a Picture (URL). After the food creation process, users should be redirected to the Home page, where the newly created food should be present.

6. Use Case 6 (Food Management)

On the Home page, logged users should see all their foods listed. Each food should have options for View, Edit, and Delete. If no foods have been created yet, a There are no foods :(message should be displayed. Users also should have the option to search for foods, based on food titles.

Foody Web App Tasks

1. Test Cases

You need to write **test cases for each of the 6 Use cases**.
Each Use case should have at **least 2 Test Cases**.
You should write **at least 15 test cases**.

2. Bug Reports

You should find and describe at least **5** bugs.
2 of the bugs should have **high ppriority** OR **blocking/critical severity**.

3. Template

Use the provided **Test-Management-and-Bug-Tracking-Template.xlsx** to document your work.

Part II - Foody API

This section provides information about the **Foody API endpoints** and their functionalities. It serves as a guide to understand how to interact with the Foody API.

Methods supported

You can see all the supported methods on the following link:

[Methods](#)

The following endpoints are supported:

1. User

- **POST /api/User/Create** – create a new user: post a JSON object in the request body, e.g. **{ "userName": "string","firstName": "string", "midName": "string", "lastName": "string", "email": "user@example.com", "password": "string", "rePassword": "string" }**

- **POST /api/User/Authentication**: log in an existing user (post a JSON object in the request body, e.g. **{ "userName": "string", "password": "string"}**)

Access Token

- When a user logs in, the response format is JSON object, e.g. **{"userName": "string", "password": "1234567", "accessToken": "eyJhbGciOiJ…"}**

NB! Access token is needed for all food requests. It should be placed under the Authorization tab, Bearer Token option.

2. Food

All of the **following requests require Authotization**.

- **GET /api/Food/All** – list all foods (empty request body)

- **GET api/Food/Search** – search foods by their name, requires **queryParameter: ?keyword=foodName**

- **POST /api/Food/Create** – create a new food: post a JSON object in the request body, e.g.

  { "name":"string", "description":"string", "url": "" }

- **PATCH /api/Food/Edit/foodId –** change the title of existing food: post a JSON object in the request body, e.g. **[{ "path": "/name", "op": "replace", "value": "string"}]**, you only have to change the value, with the new title, **leave the path and op as they are. Use Square brackets as well as curly brackets!**

Follow us:

- **DELETE** `/api/Food/Delete/foodId` – delete existing food;

Foody API Tasks: Postman Requests

Your task is to write **API requests** with Postman for certain RESTful API endpoints.
You should organize your requests in a collection.

1. Log in to the API

Send a POST request with your username and password, that you previously created from the Foody Web App. If properly executed, you will receive your access token as a part of the response body. Use it as a Bearer Token in the Authorization section for the next requests.

2. Create a new food

Send a POST request with the attributes needed for creating a new food
Keep in mind that the full attributes in the JSON body are as follows:
`{ "name": "string", description": "string", "url": "http://...jpg " }`
URL attribute is not mandatory, meaning that you can create a food only with title and description. If you decide to use the url attribute, notice, that it has to point to a picture format or it can be left blank.

3. Search for the food you created

Send a GET request to search for the food you just created by Title.

4. Change the title of the food you created

Send a PATCH request with the attributes needed for replacing the title of the food that you created

5. Delete Food

Send a DELETE request with the attributes needed for deleting the food you just patched

How to submit your work

1. Create a .zip or .rar archive containing the following:

- Test-Management-and-Bug-Tracker.xlsx file with your work;

- JSON file containing your collection of requests, created via Postman.

2. Upload your archive on the SoftUni website, into Regular Exam section.
Note: It's the same as uploading your homework.