**Test Case Planning**

| **Group members** | * Kevin Argueta * Nathan Harris * Kamarre Hicks * Mahmoud El Hajj * Justin Hackler |
| --- | --- |

1. **Use Case 1 -** Food Recognition

| **Test Case ID** | **TC01 - Successful food recognition** |
| --- | --- |
| **Test Objective** | Verify that the Food Recognition flow lets a logged-in user add a meal by camera/photo or barcode, that the system correctly identifies items, estimates portions and nutrients, allows user edits, stores the meal, and updates daily totals — meeting functional correctness, accuracy, and non-functional (performance/size/format) requirements |
| **Preconditions** | Login or signup and go to dashboard  Camera permission granted.  Storage Access Granted Reference nutrition data for the food (e.g., "apple") exists in the database and is also recognized by our Ai nutrition agent Learning. |
| **Test steps** | 1. Open app → Go to Dashboard → Click one of the 3 options (“Manually Search for Food, “Use Ai to detect Food, Scan Barcode”) 2. Take a clear photo of a Egg and upload it to our Ai scan page 3. Wait for the AI result screen. 4. Review detected item name, portion estimate, and nutrition values 5. Categorize the foods added in the results section (“lunch, “Breakfast, “Dinner”). 6. Click on the Add Meal Button Button |
| **Input Values** | * Image: Egg.jpg (JPG, 2 MB). * Image: PhillyCheaseSteak (JPG, 1MB) * Meal type: Dinner |
| **Expected results** | AI returns “2 Fried Eggs, Philly Cheesesteak with onions, white cheese and on a white hoagie” as a detected item.  Calories/macros displayed and within ±10% of reference.  The result screen appears within 10 - 60 seconds of upload.  Meal entry should be successfully saved into our database where it can be referenced and used later. |

| **Test Case ID** | **TC02 - Empty / blank image file** |
| --- | --- |
| **Test Objective** | Verify that if the user provides an empty / blank image the system recognizes it as a empty image and voids it |
| **Preconditions** | Login or signup and go to dashboard  Camera permission granted.  Storage Access Granted |
| **Test steps** | 1. Upload Blank / Empty Image 2. Wait for a result |
| **Input Values** | * Image: Empty.JPG, (JPG, 0MB) * Image: Blank.JPG (JPG, 0 MB) |
| **Expected results** | The system recognizes that it is not a real image and voids it and gives a error |

| **Test Case ID** | **TC03 - File upload that is not a image** |
| --- | --- |
| **Test Objective** | Hackers will often try to change our application by providing files that are not related to images and they may contain prompts etc. |
| **Preconditions** | Login or signup and go to dashboard  Camera permission granted.  Storage Access Granted |
| **Test steps** | 1. Upload non image file 2. Wait for the result |
| **Input Values** | * TXT File: Empty.TXT, (TXT, Under 1MB) |
| **Expected results** | The system will not accept the file |

| **Test Case ID** | **TC04 - Blurry / unreadable photo (AI low confidence)** |
| --- | --- |
| **Test Objective** | When the user uploads an image that is too blurry or too complex the system should reject the input and ask for another one |
| **Preconditions** | Login or signup and go to dashboard  Camera permission granted.  Storage Access Granted |
| **Test steps** | 1. Upload complex image 2. Wait for result |
| **Input Values** | * Image: ComplexImage.JPG (JPG, 1MB) |
| **Expected results** | The system will reject the image and say it wants another one. |