Testing Documentation

1. Use Case 1: Doctor creates a diagnosis for patient

Summary of encountered bugs:

- Bug 1: The system does not allow the doctor to add a diagnosis without selecting a patient first.

Test Cases:

- TC1: Verify that the doctor can create a diagnosis with valid inputs
- TC2: Verify that the doctor cannot create a diagnosis with invalid inputs
- TC3: Verify that the system displays an error message when the doctor tries to create a diagnosis with invalid inputs
- TC4: Verify that the doctor can view a list of all their diagnoses
- TC5: Verify that the doctor can edit a diagnosis
- TC6: Verify that the doctor can delete a diagnosis

Root causes and how they were resolved:

- Bug 1: The system did not have a validation check to ensure that a patient was selected before adding a diagnosis. This was resolved by adding a validation check to ensure that a patient is selected before adding a diagnosis.

2. Use Case 2: Patient requests a meal-plan to be generated based on diagnosis from the database

Summary of encountered bugs:

- Bug 1: The system is generating meal plans even when the diagnosis entered is not present in the database.

Test Cases:

- TC1: Verify that the patient can request a meal-plan with valid inputs
- TC2: Verify that the patient cannot request a meal-plan with invalid inputs
- TC3: Verify that the system displays an error message when the patient tries to request a meal-plan with invalid inputs
- TC4: Verify that the patient can view their meal-plan
- TC5: Verify that the patient can edit their meal-plan
- TC6: Verify that the patient can submit their meal-plan for approval by the doctor
- TC7: Verify that the doctor can view the patient's meal-plan
- TC8: Verify that the doctor can edit the patient's meal-plan
- TC9: Verify that the doctor can approve the patient's meal-plan

Root causes and how they were resolved:

- Bug 1: The system did not have a validation check to ensure that the diagnosis entered was present in the database. This was resolved by adding a validation check to ensure that the diagnosis entered is present in the database before generating a meal plan.

3. Use Case 3: Doctor is able to view, edit, and approve meal-plan of their patient

Summary of encountered bugs:

- Bug 1: The doctor is unable to view the patient's meal-plan
- Bug 2: The doctor is unable to edit the patient's meal-plan
- Bug 3: The doctor is unable to approve the patient's meal-plan
- Bug 4: The system is not displaying the correct meal plan when the doctor tries to view it.

Test Cases:

- TC1: Verify that the doctor can view the patient's meal-plan
- TC2: Verify that the doctor can edit the patient's meal-plan

- TC3: Verify that the doctor can approve the patient's meal-plan

Root causes and how they were resolved:

- Bug 1: The doctor did not have the necessary permissions to view the patient's meal-plan. This was resolved by updating the doctor's permissions.
- Bug 2: The doctor did not have the necessary permissions to edit the patient's meal-plan. This was resolved by updating the doctor's permissions.
- Bug 3: There was an issue with the approval workflow. This was resolved by updating the approval workflow and retesting the feature.
- Bug 4: The system was not retrieving the correct meal plan from the database when the doctor tries to view it. This was resolved by updating the database retrieval query to ensure that the correct meal plan is displayed.

4. Miscellaneous Front-End Test-Cases

Summary of encountered bugs:

- Bug 1: The login page is not redirecting to the user dashboard after successful login.

Test Cases:

- TC1: Enter valid login credentials and click on the login button.
- TC2: Check if the user is redirected to the dashboard page.
- TC3: Verify that the dashboard page is loaded with the user's data.

Root causes and how they were resolved:

- Bug 1: The redirection URL was not set correctly in the login API response. This was by fixing the API code to include the correct redirection URL in the response, and the issue was resolved after retesting.