Testing Documentation

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

Summary of encountered bugs:

- Bug 1: After the diagnosis is made, the patient can only see one of their past diagnosis.

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 patient can see all the diagnosis belong to themselves.

Root causes and how they were resolved:

- Bug 1: The system only fetches one instance of the diagnosis that belongs to the patient. To resolve this, fetch a list of diagnosis instances from the database that matches the patient's id.

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 generates meal plan that does not take the preference "vegetarian" into consideration.

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 falsely maps the “vegetarian” preference to “None”. To resolve it simply change the key of the meal plan mapper, from “None” to “vegetarian”.

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 approve the meal plan, but the status is not reflected on the patient side.
- Bug 2: The doctor modify the meal plan, but on the patient side it still shows the meal plan before the modification.

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: This was caused by the system fail to change the status of the instance of meal plan from the database. This is resolved simply by modifying the field status of the instance to approved and save the instance to the database.
- Bug 2: This is again caused by not changing the field of the meal plan instance. It can be resolved by modifying the text field of the instance and save the instance to the database.

4. Miscellaneous Front-End Test-Cases

Summary of encountered bugs:

- Bug 1: After user presses login the system redirect the user to the logout page.

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 html takes the user to the URL that brought the login page. In the case that the user attempt to login again right after they logout the system, they will be brought back to the logout page. Remove the “next” argument from the URL.