

# **SOFTWARE ENGINEERING**

# **TEST CASES**

# **DIET PLANNER**

**TEAM :**

# **ABDUL RAHIM – 210201097**

**INSTRUCTOR :**

# **MAM BENISH**

**TEST CASES**

Test case is a set of conditions or variables under which a tester will determine whether a system under test (SUT) works as intended. A test case typically includes specific inputs, execution conditions, and expected outcomes. The primary purpose of creating test cases is to ensure that the software behaves correctly and meets its specified requirements.

Here are key components of a test case:

**Test Case ID:** A unique identifier for the test case, often in the form of an alphanumeric code.

**Test Case Description:** A concise description of the test case, outlining the objective and the specific functionality or feature being tested.

**Module to be Tested:** Identification of the specific module or component of the software that the test case is targeting.

**Test Data:** The input data or conditions provided to the system to execute the test case.

**Test Steps:** A step-by-step sequence of actions or operations to be performed to execute the test case.

**Expected Results:** The anticipated outcomes or behavior that the tester expects from the system when the test case is executed successfully.

**Potential Results:** Possible outcomes, including pass or fail conditions, and any potential deviations or unexpected behaviors that might be observed during testing.

**Result:** The actual outcome observed when the test case is executed.

**Priority:** A classification indicating the importance or urgency of the test case, often categorized as high, medium, or low.

***Registration Component:***

**Test Case 1: Successful Registration**

**Test Case ID:** TC001

**Test Case Description:** Verify user registration functionality by entering valid user details and checking if the user is successfully registered and redirected to the login page.

**Module to be tested:** Registration Component

**Test Data:** Valid user details (name, email, password)

**Test Steps:**

* Enter valid user details in the registration form.
* Click on the registration button.

**Expected Results:** User should be successfully registered, and the system redirects to the login page.

**Potential Results:**

* **Pass:** User is successfully registered, and the system confirms the registration.
* **Fail:** Registration fails, and an error message is displayed.

**Test Case 2: Registration with Existing Email**

**Test Case ID:** TC002

**Test Case Description:** Verify that entering user details with an existing email results in an appropriate error message.

**Module to be tested:** Registration Component

**Test Data:** User details with an email that already exists in the database

**Test Steps:**

* Enter user details with an existing email.
* Click on the registration button.

**Expected Results:** An appropriate error message should be displayed, indicating that the email is already in use.

**Potential Results:**

* **Pass:** Error message is displayed.
* **Fail:** No error message or incorrect error message is displayed.

**Test Case 3: Registration with Invalid Password**

**Test Case ID:** TC003

**Test Case Description:** Verify that entering valid user details with an invalid password (e.g., less than 8 characters) results in an appropriate error message.

**Module to be tested:** Registration Component

**Test Data:** Valid user details with an invalid password

**Test Steps:**

* Enter valid user details with an invalid password.
* Click on the registration button.

**Expected Results:** An error message should be displayed, indicating the password requirements.

**Potential Results:**

* **Pass:** Error message is displayed.
* **Fail:** No error message or incorrect error message is displayed.

**Test Case 4: Blank Fields in Registration**

**Test Case ID:** TC004

**Test Case Description:** Verify that leaving some mandatory fields blank in the registration form results in appropriate error messages.

**Module to be tested:** Registration Component

**Test Data:** Leaving some mandatory fields blank in the registration form

**Test Steps:**

* Leave some mandatory fields blank in the registration form.
* Click on the registration button.

**Expected Results:** Appropriate error messages should be displayed for the blank fields.

**Potential Results:**

* **Pass:** Error messages are displayed for blank fields.
* **Fail:** No error messages or incorrect error messages are displayed.

***Login Component:***

**Test Case 5: Successful Login**

**Test Case ID:** TC005

**Test Case Description:** Verify successful login functionality by entering valid credentials and checking if the user is directed to the menu tab.

**Module to be tested:** Login Component

**Test Data:** Valid login credentials

**Test Steps:**

* Enter valid credentials in the login form.
* Click on the login button.

**Expected Results:** User should be successfully logged in and directed to the menu tab.

**Potential Results:**

* **Pass:** User is successfully logged in.
* **Fail:** Login fails, and an error message is displayed.

**Test Case 6: Login with Invalid Credentials**

**Test Case ID:** TC006

**Test Case Description:** Verify that entering invalid credentials results in an appropriate error message.

**Module to be tested:** Login Component

**Test Data:** Invalid login credentials

**Test Steps:**

* Enter invalid credentials in the login form.
* Click on the login button.

**Expected Results:** An error message should be displayed, indicating incorrect login details.

**Potential Results:**

* **Pass:** Error message is displayed.
* **Fail:** No error message or incorrect error message is displayed.

***Calculate Obesity Component:***

**Test Case 7: Calculate Obesity with Normal BMI**

**Test Case ID:** TC007

**Test Case Description:** Verify that entering weight and age within normal BMI range results in displaying "Maintained" as the obesity status.

**Module to be tested:** Calculate Obesity Component

**Test Data:** Weight and age within normal BMI range

**Test Steps:**

* Enter weight and age within normal BMI range.
* Click on the calculate button.

**Expected Results:** The system should display "Maintained" as the obesity status.

**Potential Results:**

* **Pass:** "Maintained" is displayed.
* **Fail:** Incorrect status or no status is displayed.

**Test Case 8: Calculate Obesity with Overweight BMI**

**Test Case ID:** TC008

**Test Case Description:** Verify that entering weight and age within overweight BMI range results in displaying "Overweight" as the obesity status.

**Module to be tested:** Calculate Obesity Component

**Test Data:** Weight and age within overweight BMI range

**Test Steps:**

* Enter weight and age within overweight BMI range.
* Click on the calculate button.

**Expected Results:** The system should display "Overweight" as the obesity status.

**Potential Results:**

* **Pass:** "Overweight" is displayed.
* **Fail:** Incorrect status or no status is displayed.

**Test Case 9: Calculate Obesity with Underweight BMI**

**Test Case ID:** TC009

**Test Case Description:** Verify that entering weight and age within underweight BMI range results in displaying "Underweight" as the obesity status.

**Module to be tested:** Calculate Obesity Component

**Test Data:** Weight and age within underweight BMI range

**Test Steps:**

* Enter weight and age within underweight BMI range.
* Click on the calculate button.

**Expected Results:** The system should display "Underweight" as the obesity status.

**Potential Results:**

* **Pass:** "Underweight" is displayed.
* **Fail:** Incorrect status or no status is displayed.

***Daily Chronometer Component:***

**Test Case 10: Calculate Calories with Valid Food Items**

**Test Case ID:** TC010

**Test Case Description:** Verify that entering different types and quantities of valid food items results in displaying the correct total calorie count.

**Module to be tested:** Daily Chronometer Component

**Test Data:** Different types and quantities of valid food items

**Test Steps:**

* Enter different types and quantities of valid food items.
* Click on the calculate button.

**Expected Results:** The system should display the correct total calorie count.

**Potential Results:**

* **Pass:** Correct total calorie count is displayed.
* **Fail:** Incorrect or no calorie count is displayed.

**Test Case 11: Calculate Calories with Invalid Quantity**

**Test Case ID:** TC011

**Test Case Description:** Verify that entering food items with invalid quantities (e.g., alphabets) results in an appropriate error message.

**Module to be tested:** Daily Chronometer Component

**Test Data:** Food items with invalid quantities (e.g., alphabets)

**Test Steps:**

* Enter food items with invalid quantities.
* Click on the calculate button.

**Expected Results:** The system should not accept invalid quantities and should display an appropriate error message.

**Potential Results:**

* **Pass:** Error message is displayed.
* **Fail:** No error message or incorrect error message is displayed.

***Diet Plan Component:***

**Test Case 12: Access Diet Plan for Each Day**

**Test Case ID:** TC012

**Test Case Description:** Verify that accessing the diet plan for each day results in displaying the correct plan.

**Module to be tested:** Diet Plan Component

**Test Steps:**

* Access the diet plan for each day.

**Expected Results:** The system should display the correct diet plan for each day.

**Potential Results:**

* **Pass:** Correct diet plan is displayed.
* **Fail:** Incorrect or no diet plan is displayed.

**Test Case 13: Access Diet Plan with Incorrect Data Type**

**Test Case ID:** TC013

**Test Case Description:** Verify that entering incorrect data types (e.g., alphabets) while accessing the diet plan results in an appropriate error message.

**Module to be tested:** Diet Plan Component

**Test Steps:**

* Enter incorrect data types while accessing the diet plan.

**Expected Results:** The system should not accept incorrect data types and should display an appropriate error message.

**Potential Results:**

* **Pass:** Error message is displayed.
* **Fail:** No error message or incorrect error message is displayed.

These test cases cover various scenarios to ensure the proper functionality and robustness of "Diet Planner" application.

***THE END***