

# **SOFTWARE ENGINEERING**

# **TEST CASES**

# **DIET PLANNER**

**TEAM :**

# **ABDUL RAHIM – 210201097**

**INSTRUCTOR :**

# **MAM BENISH**

UNIT / SYSTEM TESTING :

Unit testing is a software testing technique in which individual units or components of a software application are tested in isolation from the rest of the system. The main purpose of unit testing is to validate that each unit of the software performs as designed. A unit is the smallest testable part of an application, typically a function or method.

System testing involves testing the complete and integrated software system as a whole. It tests the system's compliance with specified requirements and verifies that it meets the intended business objectives.

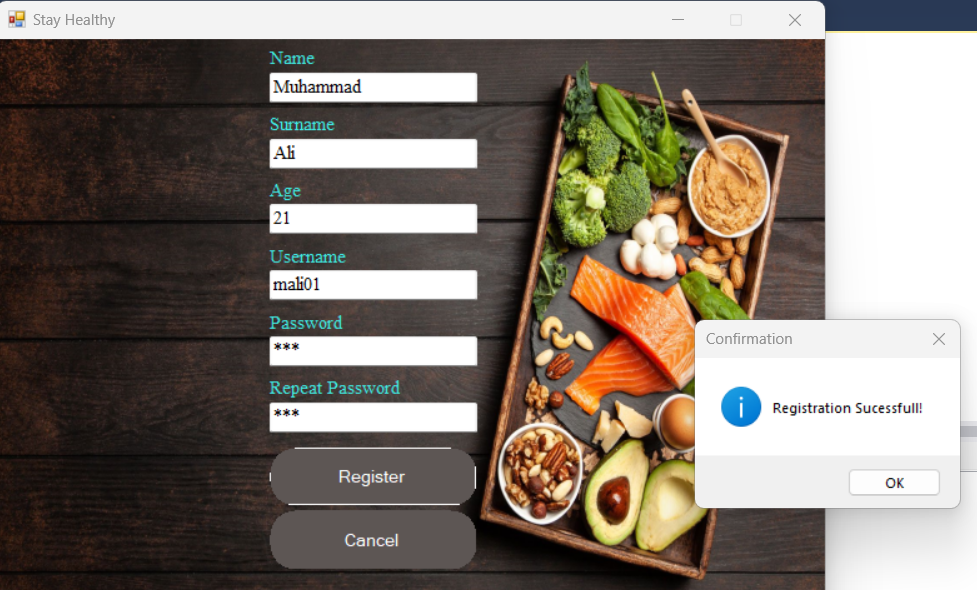
**UNIT**

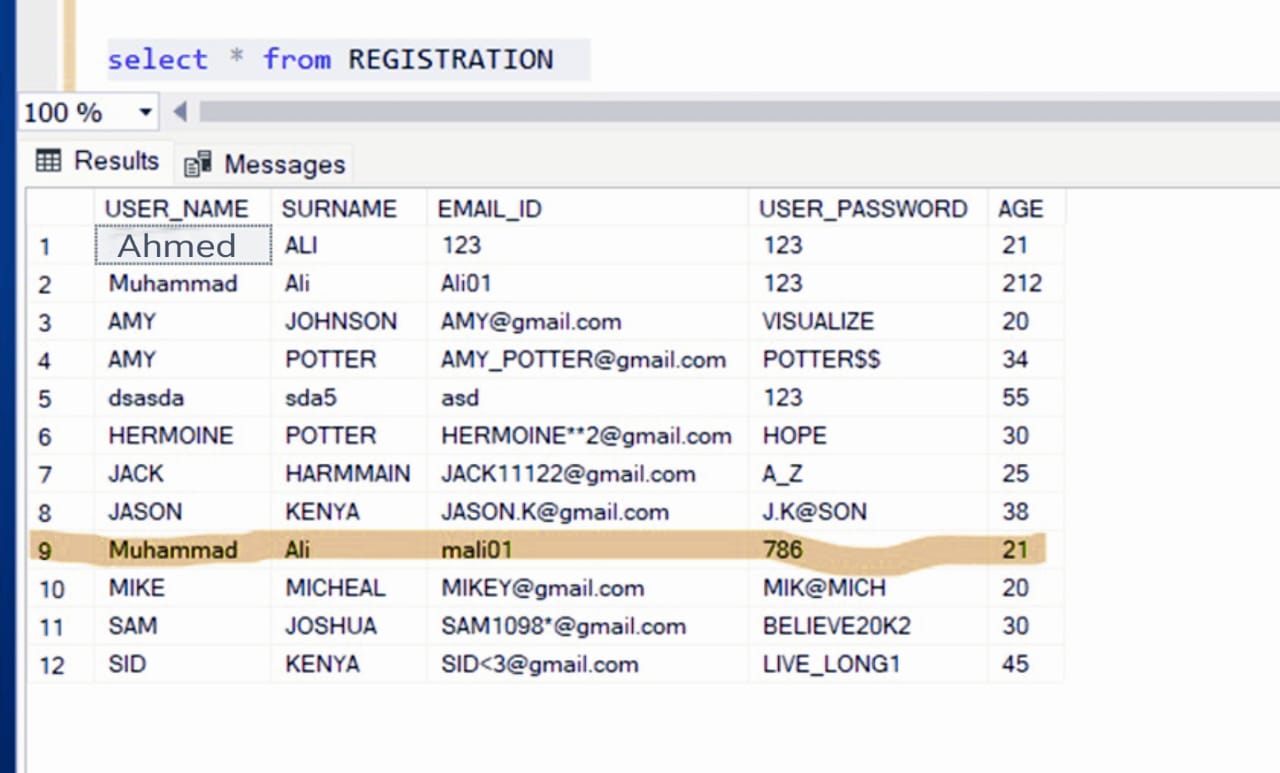
* **REGISTRATION COMPONENT**

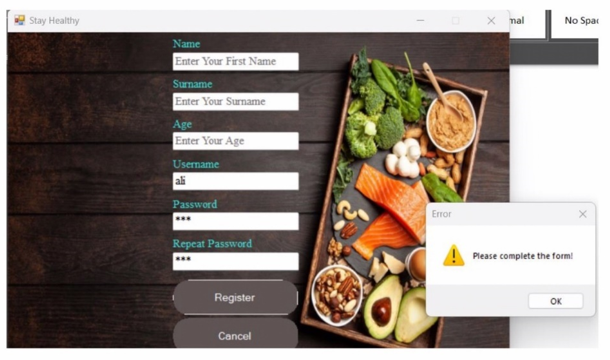
We have test the registration component by entering various combinations of valid and invalid information and verify that the user's information is correctly saved in the database and error messages are displayed appropriately.

**SYSTEM**

* Verifying that the login page loads correctly and displays the login and registration buttons.
* Testing the registration process by creating a new account and verifying that the user's information is correctly saved in the database.
* Testing the app's error by entering incorrect or incomplete inputs and verifying that appropriate messages are displayed.







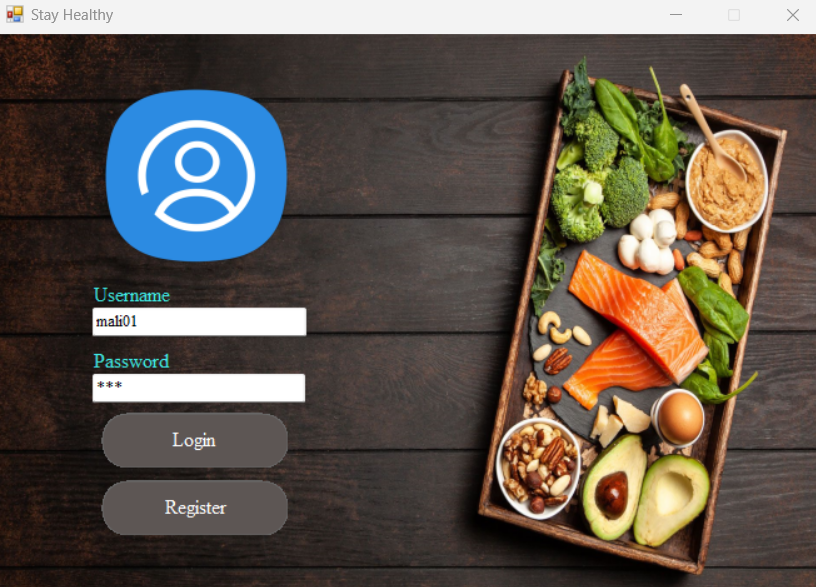
**UNIT**

* **LOGIN COMPONENT**

We have test the login component by logging in with various combinations of valid and invalid credentials and verify that the user is correctly directed to the appropriate page (e.g. menu tab or error message).

**SYSTEM**

* Testing the login process by logging in with the newly created account and verifying that the user is taken to the menu tab.



A menu on a table

Description automatically generated

**UNIT**

* **CALCULATE OBESITY COMPONENT**

We have test the Calculate Obesity component by entering various combinations of weight and age and verify that the correct obesity status (maintained, overweight, or underweight) is displayed and the appropriate diet plan is generated.

**SYSTEM**

* Testing the Calculate Obesity feature by entering various combinations of weight and age and verifying that the correct obesity status (maintained, overweight, or underweight) is displayed.

A person measuring their waist

Description automatically generated

A screenshot of a computer

Description automatically generated

**UNIT**

* **DAILY CHRONOMETER COMPONENT**

We have test the Daily Chronometer component by entering different types and quantities of food items and verify that the correct total calorie count is displayed.

**SYSTEM**

* Testing the Daily Chronometer by choosing different types and entering quantities of food items and verifying that the correct total calorie count is displayed.

A screenshot of a calorie counter

Description automatically generated

**UNIT**

* **DIET PLAN COMPONENT**

Test the diet plan component by accessing the diet plan for each day and verifying that the correct plan is displayed.

**SYSTEM**

* Test the Your Diet Plan feature by accessing the diet plan for each day and verifying that the correct plan is displayed.

A screenshot of a diet plan

Description automatically generated

• We also tested it by entering incorrect data type in different places, like we have enter quantity in alphabets in Daily Chronometer and make sure that it will not take it.