**Nutrition and Diet Application Project Test Plan**

Rohandra Macolm

Dobrian Zaprianov

Diamond Zetty

Dorrell Zimmerman

**Table of Contents**

Introduction: Provides a brief overview of the software project and its testing scope…………………………………………………………………………………………………………………....3

Testing Objectives: Clearly defines the objectives of the testing effort, including the quality attributes to be assessed…………………………………………………………………………3

Test Strategy: Outlines the overall approach to testing, including the types of testing to be performed and the testing methodologies to be employed……………………….3

Test Scope: Defines the boundaries and limitations of the testing effort, including what is included and excluded…………………………………………………………………………………3

Test Cases: Provides detailed descriptions of the test cases designed to cover various aspects of the software's functionality, performance, and usability………………….4

Testing Procedures: Documents the specific steps and procedures for executing the test cases, including test data preparation and test environment configuration………7

Testing Schedule: Develops a detailed testing schedule with deadlines for each testing phase………………………………………………………………………………………………………….8

Testing Resources: Identifies and allocates the necessary resources for the testing effort, including personnel, equipment, and budget…………………………………………………..8

Defect Management: Establishes a process for identifying, reporting, and tracking defects or bugs encountered during testing………………………………………………….8

Appendix A: Test Case Figures………………………………………………………………………9

**Introduction:**

The Nutrition and Diet Application Project purpose is to help individuals reach their weight goals by monitoring daily calorie intake. It will calculate the Body Mass Index (BMI) when provided with height and weight of the individual. It will determine the Basal Metabolic Rate (BMR) from height, weight, age, and gender. Finally, it will keep a food diary to track personal daily calorie intakes. The testing scope of this application includes boundaries, deliverables, objectives, and criteria of the project.

**Testing objectives:**

The main objective of the testing effort is to design a robust and bug free application. The quality attributes that will be assessed include input validation, boundary protection and crash resistance. Input validation will be tested by subjecting the web application to non-numerical data, non-integer numbers. However, for the food item the application must allow only letters! The boundary protection must satisfy rejection of negative numbers and the number zero. The deliverables are accurately calculated BMI, BMR indexes. Correct and meaningful Food Dairy table. The software application must avoid a division by zero crash from intentional zero number input.

**Test strategy:**

Main test strategy is focused on user interaction with the software. The GUI must perform the required operations. Input validation must reject any illegal entries. All application calculations must be valid. Finally, the output must be clear and concise.

**Test scope:**

The application boundaries will be defined by accepting only positive numbers excluding zero. Any letters or special characters must be rejected. Only integer values must be allowed. For the food item only, alphabetical input must be allowed. The limitation of the project test is that the upper boundaries of the numbers will not be verified and will be excluded from the test plan.

**Test cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case #  And figure # | Date | Action | Expected  result | Actual  result | Result and Severity: |
| 1 | 06/10/24 | Double click on application icon | GUI menu displays | GUI menu displays | Pass |
| 2 | 06/10/24 | Mouse click on Calculators menu | Drop down menu displays | Drop down menu displays | Pass |
| 3 | 06/10/24 | Mouse click on Food Diary | Daily Food Diary page displays | Daily Food Diary page displays | Pass |
| 4 | 06/10/24 | Mouse click on Home Page | Home Page displays | Home Page displays | Pass |
| 5 | 06/10/24 | Mouse click on BMI calculator | BMI calculator page shows | BMI calculator page shows | Pass |
| 6 | 06/10/24 | Mouse click on BMR calculator | BMR calculator page shows | BMR calculator page shows | Pass |
| 7 | 06/11/24 | On BMI calculator page: enter feet, inches and weight then click on Calculate BMI | BMI is calculated and showing | BMI is calculated and showing; however the output is not properly formatted and input data is erased! | Fail  Low severity  Problem |
| 8 | 06/11/24 | On BMI calculator page: enter feet, inches and weight.  In real numbers then click on Calculate BMI | Error message prompts to enter an integer value | Error message prompts to enter an integer value | Pass |
| 9 | 06/11/24 | On BMI calculator page: enter feet, inches and weight.  As negative numbers then click on Calculate BMI | Displays an error message | Shows a negative result | Fail  Medium Severity problem |
| 10 | 06/11/24 | On BMI calculator page: enter feet, inches and weight.  As zero then click on Calculate BMI | Displays an error message | Results in an app crash  Because of division by zero! | Fail  High severity problem |
| 11 | 06/12/24 | On BMR calculator page: enter height, weight, and age  then click on Calculate BMR | BMR is calculated and showing | BMR is calculated and showing; however the output is not properly formatted and input data is erased! | Fail  Low severity problem |
| 12 | 06/12/24 | On BMR calculator page: enter height, weight, and age in real numbers then click on Calculate BMR | Error message prompts to enter an integer value | Error message prompts to enter an integer value | Pass |
| 13 | 06/12/24 | On BMR calculator page: enter height, weight, and age as negative numbers then click on Calculate BMR | Displays an error message | Shows a negative result! | Fail  Medium severity problem |
| 14 | 06/12/24 | On BMR calculator page: enter height, weight, and age as zero values then click on Calculate BMR | Displays an error message | Results in invalid output | Fail  Medium severity problem |
| 15 | 06/13/24 | On the food diary page for food item enter: egg and for calories enter: 70 | Displays the food item in the list | Displays the food item in the list | Pass |
| 16 | 06/13/24 | On the food diary page for food item enter: egg and for calories enter: -70 | Displays an error message | Displays the food item in the list | Fail  Medium severity problem |
| 17 | 06/13/24 | On the food diary page:  For food item enter: numbers and special caracters | Displays an error message | Displays the food item in the list | Fail  Medium severity problem |
| 18 | 6/13/24 | On the food diary page for food item enter: egg and for calories enter: 70 and bread with 100 calories | Displays properly multiple items in the list and total calories | Displays properly multiple items in the list and total calories | Pass |
|  |  |  |  |  |  |

**Testing Procedures:**

The test data preparation includes all data types! The test environment is:

Computer:

Device name LAPTOP-CU95MKVJ

Processor 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz

Installed RAM 8.00 GB (7.77 GB usable)

Device ID 2B1462B0-798A-4468-89AD-5165E013B4E5

Product ID 00325-82029-05452-AAOEM

System type 64-bit operating system, x64-based processor

Operating System:

Edition Windows 11 Home

Version 23H2

Installed on ‎12/‎23/‎2022

OS build 22631.3672

Experience Windows Feature Experience Pack 1000.22700.1009.0

Environment:

PyCharm 2024.1.2 (Community Edition)

Build #PC-241.17011.127, built on May 28, 2024

Runtime version: 17.0.11+1-b1207.24 amd64

VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.

Windows 11.0

GC: G1 Young Generation, G1 Old Generation

Memory: 2048M

Cores: 8

**Testing schedule:**

|  |  |  |
| --- | --- | --- |
| Test Case # | Date | Test Type |
| 1 through 6 | 6/10/2024 | Test page GUI and connectivity |
| 7 through 10 | 6/11/2024 | Testing BMI calculations and input validation |
| 11 through 14 | 6/12/2024 | Testing BMR calculations and input validation |
| 15 though 17 | 6/13/2024 | Testing Food Diary page functionality and input validation |
|  |  |  |

**Testing Resources:**

Personnel:

Rohandra Macolm: back-end developer

Dobrian Zaprianov: product test developer and user-guide designer

Diamond Zetty: front-end developer

Dorrell Zimmerman: project manager and documentation designer

Equipment:

Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz

Installed RAM: 8.00 GB (7.77 GB usable)

Budget:

The computer cost is: $2000. The salary is $50 per hour and the daily is $400 per software developer. Project testing required a total of four days. The testing budget is therefore: $2000 computer + 4 developers \* 4 days \* 400 per day = $8400 total cost!

**Defect management:**

Identifying bugs and defects is done by assessing all aspects of the project including web page connectivity, calculations correctness and input validation.

Reporting bugs to the front and back-end developers using Skype and GitHub.

Tracking defects by doing multiple types of application testing. Organize and attend weekly meetings to assess and mitigate project issues.

**Appendix A**

A screenshot of a computer

Description automatically generated

Figure 1

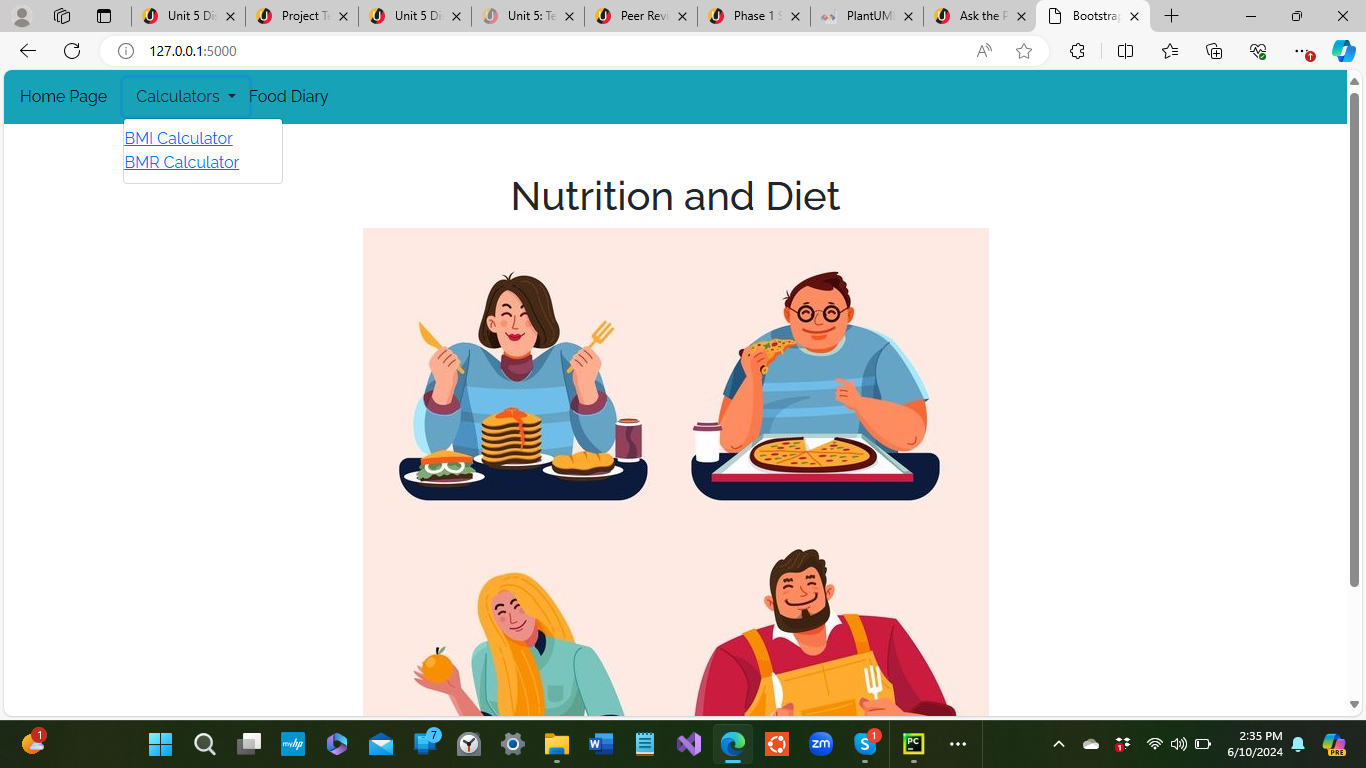
Figure 2

Figure 3A screenshot of a computer

Description automatically generated

Figure 4A screenshot of a computer

Description automatically generated

Figure 5A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure 6A screenshot of a computer

Description automatically generated

Figure 7A screenshot of a computer

Description automatically generated

Figure 8A screenshot of a computer

Description automatically generated

Figure 9A screenshot of a computer

Description automatically generated

Figure 10A screenshot of a computer

Description automatically generated

Figure 11A screenshot of a computer

Description automatically generated

Figure 12A computer screen with a white and blue box

Description automatically generated

Figure 13A screenshot of a computer

Description automatically generated

Figure 14A screenshot of a computer

Description automatically generated

Figure 15A screenshot of a computer

Description automatically generated

Figure 16A screenshot of a computer

Description automatically generated

Figure 17A screenshot of a computer

Description automatically generated

Figure 18