

# Verification and Validation Report: REVITALIZE

Team 13,  
Bill Nguyen  
Syed Bokhari  
Hasan Kibria  
Mahmoud Anklis  
Youssef Dahab  
Logan Brown

March 9, 2023

# 1 Revision History

Date	Version	Notes
March 5th, 2023	Bill Nguyen	Adding Unit Tests for Workout and Rest Section
March 5th, 2023	Youssef Dahab	Added Functional Requirements Evalua- tion
March 6th, 2023	Youssef Dahab	Added Changes Due To Testing
March 8th, 2023	Hasan Kibria	Adding Unit Tests for Diet Section
March 8th, 2023	Youssef Dahab	Added Reflection

## 2 Symbols, Abbreviations and Acronyms

symbol	description
REVITALIZE	Name of application
SRS	Software Requirements Specification
VnV	Verification and Validation
FR	Functional Requirement
NFR	Non Functional Requirement
LP	Login Page
SP	Sign-up Page
MP	Main Page or Maintainability and Portability Requirements
DS	Diet Section
WS	Workout Section
RS	Rest Section
LF	Look and Feel Requirements
UH	Usability and Humanity Requirements
PE	Performance Requirement
OE	Operational Requirement
SE	Security Requirement
CU	Cultural Requirement

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>Functional Requirements Evaluation</b>	<b>1</b>
3.1	Login Page . . . . .	1
3.2	Signup Page . . . . .	5
3.3	Main Page . . . . .	7
3.4	Diet Section Page . . . . .	9
3.5	Workout Section Page . . . . .	13
3.6	Rest Section Page . . . . .	16
<b>4</b>	<b>Nonfunctional Requirements Evaluation</b>	<b>17</b>
4.1	Usability . . . . .	17
4.2	Performance . . . . .	17
4.3	etc. . . . .	17
<b>5</b>	<b>Comparison to Existing Implementation</b>	<b>17</b>
<b>6</b>	<b>Unit Testing</b>	<b>17</b>
6.1	Workout Section . . . . .	17
6.2	Rest Section . . . . .	20
6.3	Diet Section . . . . .	22
<b>7</b>	<b>Changes Due to Testing</b>	<b>23</b>
<b>8</b>	<b>Automated Testing</b>	<b>23</b>
<b>9</b>	<b>Trace to Requirements</b>	<b>23</b>
<b>10</b>	<b>Trace to Modules</b>	<b>28</b>
<b>11</b>	<b>Code Coverage Metrics</b>	<b>29</b>
<b>12</b>	<b>Reflection Appendix</b>	<b>29</b>

## List of Tables

1	Workout Section Unit Tests Part 1 . . . . .	18
2	Workout Section Unit Tests Part 2 . . . . .	19
3	Rest Section Unit Tests Part 1 . . . . .	20
4	Rest Section Unit Tests Part 2 . . . . .	21
5	Diet Section Unit Tests Part 1 . . . . .	22
6	<b>Traceability Matrix for Login Page Functional Requirements</b> . . . . .	23
7	<b>Traceability Matrix for Signup Page Functional Requirements</b> . . . . .	24
8	<b>Traceability Matrix for Main Page Functional Requirements</b> . . . . .	24
9	<b>Traceability Matrix for Diet Page Functional Requirements</b> . . . . .	25
10	<b>Traceability Matrix for Workout Page Functional Requirements</b> . . . . .	25
11	<b>Traceability Matrix for Rest Section Functional Requirements</b> . . . . .	25
12	<b>Traceability Matrix for Look and Feel Nonfunctional Requirements</b> . . . . .	26
13	<b>Traceability Matrix for Usability and Humanity Nonfunctional Requirements</b> . . . . .	26
14	<b>Traceability Matrix for Performance Nonfunctional Requirements</b> . . . . .	26
15	<b>Traceability Matrix for Operational Nonfunctional Requirements</b> . . . . .	27
16	<b>Traceability Matrix for Maintainability and Portability Nonfunctional Requirements</b> . . . . .	27
17	<b>Traceability Matrix for Security Nonfunctional Requirements</b> . . . . .	27
18	<b>Traceability Matrix for Cultural and Political Nonfunctional Requirements</b> . . . . .	27
19	<b>Trace Between Requirements and Modules</b> . . . . .	28

## List of Figures

This document details the complete testing process for REVITALIZE, as laid out in the project test plan. It contains an evaluation of the project's functional and non-functional requirements that are defined in the **SRS**, the changes made due to testing, and an analysis of the traceability between requirements and modules.

## 3 Functional Requirements Evaluation

### 3.1 Login Page

<b>Test #1:</b>	<b>FR-LP-1</b>
<b>Description:</b>	Testing that login page is displayed upon starting the application
<b>Type:</b>	Manual
<b>Initial State:</b>	Loading stage of the login page
<b>Input:</b>	An event that loads the login page
<b>Output:</b>	Login page is displayed with all necessary components
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #2:</b>	<b>FR-LP-2</b>
<b>Description:</b>	Testing that login page displays fillable username textbox
<b>Type:</b>	Manual
<b>Initial State:</b>	Login page is displayed with username textbox
<b>Input:</b>	Enter username information in textbox
<b>Output:</b>	Username information entered is displayed in textbox
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #3:</b>	<b>FR-LP-3</b>
<b>Description:</b>	Testing that login page displays fillable password textbox
<b>Type:</b>	Manual
<b>Initial State:</b>	Login page is displayed with password textbox
<b>Input:</b>	Enter password information in textbox
<b>Output:</b>	Password information entered is displayed in textbox via hidden text
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #4:</b>	<b>FR-LP-4</b>
<b>Description:</b>	Testing that login page displays login button
<b>Type:</b>	Manual
<b>Initial State:</b>	Login page is displayed with login button
<b>Input:</b>	Click the login button
<b>Output:</b>	User logs in after the system checks the validity of the input parameters in the login page
<b>Expected:</b>	Login button is displayed and user logged in successfully
<b>Result:</b>	PASS

<b>Test #5:</b>	<b>FR-LP-5</b>
<b>Description:</b>	Testing that login page displays forgot password button
<b>Type:</b>	Manual
<b>Initial State:</b>	Login page is displayed with forgot password button
<b>Input:</b>	Click forgot password button
<b>Output:</b>	Display forgot password screen with textbox to enter email
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #6:</b>	<b>FR-LP-6</b>
<b>Description:</b>	Testing that login page displays a stay logged in checkbox
<b>Type:</b>	Manual
<b>Initial State:</b>	Login page is displayed with stay logged in checkbox that is empty
<b>Input:</b>	Click stay logged in checkbox
<b>Output:</b>	Display a check-mark in the stay logged in checkbox if checkbox is empty. Else if checkbox contains check-mark already it will then display an empty checkbox
<b>Expected:</b>	
<b>Result:</b>	PASS



<b>Test #7:</b>	<b>FR-LP-7</b>
<b>Description:</b>	Testing that application saves prior login information if stay logged in checkbox is checked
<b>Type:</b>	Manual
<b>Initial State:</b>	Loading stage of REVITALIZE where previous state had stay logged in checkbox checked
<b>Input:</b>	An event that loads REVITALIZE
<b>Output:</b>	Display main page , with same data from previous state of main page
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #8:</b>	<b>FR-LP-8</b>
<b>Description:</b>	Testing that login page displays sign-up button that redirects to sign-up page
<b>Type:</b>	Manual
<b>Initial State:</b>	Login page is displayed with sign up button
<b>Input:</b>	Click sign up button
<b>Output:</b>	Loads and displays sign up page
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #9:</b>	<b>FR-LP-9</b>
<b>Description:</b>	Testing if application checks validity of input parameters in login page
<b>Type:</b>	Manual
<b>Initial State:</b>	Login page is displayed with inputted information in username and password text-boxes
<b>Input:</b>	Click login button
<b>Output:</b>	If failure state, display an invalid password or username banner. Else if success state, load and display main page
<b>Expected:</b>	
<b>Result:</b>	PASS

### 3.2 Signup Page

<b>Test #10:</b>	<b>FR-SP-1</b>
<b>Description:</b>	Testing that signup page displays fillable username textbox
<b>Type:</b>	Manual
<b>Initial State:</b>	Signup page is displayed with username textbox
<b>Input:</b>	Enter username information in textbox
<b>Output:</b>	Username information entered is displayed in textbox
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #11:</b>	<b>FR-SP-2</b>
<b>Description:</b>	Testing that signup page displays fillable password textbox
<b>Type:</b>	Manual
<b>Initial State:</b>	Signup page is displayed with password textbox
<b>Input:</b>	Enter password information in textbox
<b>Output:</b>	Password information entered is displayed in textbox via hidden text
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #12:</b>	<b>FR-SP-3</b>
<b>Description:</b>	Testing that signup page displays fillable email textbox
<b>Type:</b>	Manual
<b>Initial State:</b>	Signup page is displayed with email textbox
<b>Input:</b>	Enter email information in textbox
<b>Output:</b>	Email information entered is displayed in textbox
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #13:</b>	<b>FR-SP-4</b>
<b>Description:</b>	Testing that signup page displays signup button
<b>Type:</b>	Manual
<b>Initial State:</b>	Signup page is displayed with signup button
<b>Input:</b>	Click the signup button
<b>Output:</b>	User signs up after the system checks the validity of the input parameters on the signup page
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #14:</b>	<b>FR-SP-5</b>
<b>Description:</b>	Testing if application checks validity of input parameters in signup page
<b>Type:</b>	Manual
<b>Initial State:</b>	Signup page is displayed with inputted information in username and password text-boxes
<b>Input:</b>	Click signup button
<b>Output:</b>	If failure state then display an invalid username/password or email banner. Else if success state then load and display login page
<b>Expected:</b>	
<b>Result:</b>	PASS

### 3.3 Main Page

<b>Test #15:</b>	<b>FR-MP-1</b>
<b>Description:</b>	Testing that the application displays a calendar with current date on successful login
<b>Type:</b>	Manual
<b>Initial State:</b>	Main page is displayed with calendar of current date
<b>Input:</b>	An event that loads the main page
<b>Output:</b>	Main page is displayed with all necessary components
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #16:</b>	<b>FR-MP-2</b>
<b>Description:</b>	Testing that the application has a previous day and a next day button on each page after successful login
<b>Type:</b>	Manual
<b>Initial State:</b>	Main page and Diet, Workout, Rest sections are displayed with previous day and next day buttons
<b>Input:</b>	An event that loads the main page, Diet, Workout, Rest sections and the previous day and next day buttons are clicked
<b>Output:</b>	Main page, Diet, Workout, Rest sections are displayed with previous day and next day buttons. Once the next day button is clicked, the calendar refreshes the calendar information for the next day. Once the previous day button is clicked, the calendar refreshes the calendar information for the previous day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #17:</b>	<b>FR-MP-3</b>
<b>Description:</b>	Testing that a back button is displayed on each user interface after a section is selected
<b>Type:</b>	Manual
<b>Initial State:</b>	Each interaction after leaving the main page must have a visible back button
<b>Input:</b>	An event that loads the next user interface after leaving the main page and the back button is clicked
<b>Output:</b>	The next user interface after leaving the main page is displayed with a back button. Once the back button is clicked the main page is loaded
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #18:</b>	<b>FR-MP-4</b>
<b>Description:</b>	Testing that the application displays the sections Diet, Exercise, and Rest on the current calendar day
<b>Type:</b>	Manual
<b>Initial State:</b>	Main page is displayed with Diet, Exercise and Rest buttons available to click
<b>Input:</b>	An event that loads the main page and the Diet, Exercise and Rest buttons are clicked
<b>Output:</b>	Main page is displayed with Diet, Exercise and Rest buttons. If the Diet button is clicked, the Diet interface is loaded. If the Exercise button is clicked, the Exercise interface is loaded. If the Rest button is clicked, the Rest interface is loaded
<b>Expected:</b>	
<b>Result:</b>	PASS

### 3.4 Diet Section Page

<b>Test #19:</b>	<b>FR-DS-1</b>
<b>Description:</b>	Testing that application prompts the user to height, input dietary, weight, calorie information on initial launch of Diet section
<b>Type:</b>	Manual
<b>Initial State:</b>	Diet section is initialized for the first time and an initial information dialog is launched
<b>Input:</b>	An event that loads the diet section for the first time
<b>Output:</b>	A fillable dialog box is launched with height, dietary information, weight and calorie information
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #20:</b>	<b>FR-DS-2</b>
<b>Description:</b>	Testing that the application saves initial user height, dietary, weight, calorie information
<b>Type:</b>	Manual
<b>Initial State:</b>	Diet section is initialized for the first time and an initial information dialog is launched
<b>Input:</b>	Initial information dialog values are filled
<b>Output:</b>	Initial information values are saved to the database
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #21:</b>	<b>FR-DS-3</b>
<b>Description:</b>	Testing that the application initializes with a list of food logged on the current calendar day
<b>Type:</b>	Manual
<b>Initial State:</b>	Section is initialized with a list of food logged for the current calendar day
<b>Input:</b>	An event that loads the rest section
<b>Output:</b>	A list of inputted food is loaded for the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #22:</b>	<b>FR-DS-4</b>
<b>Description:</b>	Testing that Diet section displays add food button
<b>Type:</b>	Manual
<b>Initial State:</b>	Diet section is displayed with add food button
<b>Input:</b>	Click add food button
<b>Output:</b>	A user interface is launched that lets the user select between searching for food or adding a custom meal
<b>Expected:</b>	
<b>Result:</b>	PASS



<b>Test #23:</b>	<b>FR-DS-5</b>
<b>Description:</b>	Testing that Diet section displays search food button
<b>Type:</b>	Manual
<b>Initial State:</b>	Food adding user interface is displayed with search food button
<b>Input:</b>	Click the search food button
<b>Output:</b>	A recipe criteria user interface is launched that displays a list of modifiable criteria and a search button
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #24:</b>	<b>FR-DS-6</b>
<b>Description:</b>	Testing that search food button launches recipe criteria user interface
<b>Type:</b>	Manual
<b>Initial State:</b>	Recipe criteria user interface is launched
<b>Input:</b>	Search criteria is modified and search button is clicked
<b>Output:</b>	List of recipes are loaded correctly based on constraints of search criteria
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #25:</b>	<b>FR-DS-7</b>
<b>Description:</b>	Testing that recipe search displays correct recipe values based on input constraints
<b>Type:</b>	Manual
<b>Initial State:</b>	Recipe list is loaded based on search constraints
<b>Input:</b>	Add recipe button is clicked
<b>Output:</b>	Selected recipe is added to the list of food logged on the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #26:</b>	<b>FR-DS-8</b>
<b>Description:</b>	Testing that add custom meal button adds meal to list of logged food for the current calendar day upon filling necessary recipe information textboxes
<b>Type:</b>	Manual
<b>Initial State:</b>	Food adding interface is displayed with add custom meal button
<b>Input:</b>	Click add custom meal button
<b>Output:</b>	A dialog box is launched that lets the user fill in custom meal information. The meal is added to the food log list of the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

### 3.5 Workout Section Page

<b>Test #27:</b>	<b>FR-WS-1</b>
<b>Description:</b>	Testing that the Workout section initializes with a preset list of exercises on the current calendar day
<b>Type:</b>	Manual
<b>Initial State:</b>	Workout section is initialized with a preset list of exercises of the current calendar day
<b>Input:</b>	An event that loads the workout section
<b>Output:</b>	A preset list of exercises is loaded for the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #28:</b>	<b>FR-WS-2</b>
<b>Description:</b>	Testing that the Workout section has add exercise button
<b>Type:</b>	Manual
<b>Initial State:</b>	Workout section is displayed with add exercise button
<b>Input:</b>	Click add exercise button
<b>Output:</b>	A dialog box is launched that lets the user ll custom exercise information. The exercise is added to the exercise list of the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #29:</b>	<b>FR-WS-3</b>
<b>Description:</b>	Testing that the Workout section has delete exercise button
<b>Type:</b>	Manual
<b>Initial State:</b>	Each exercise in the workout section is displayed with a delete exercise button
<b>Input:</b>	Click delete exercise button
<b>Output:</b>	The exercise is deleted from the exercise list of the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #30:</b>	<b>FR-WS-4</b>
<b>Description:</b>	Testing that exercises display an edit exercise button that launches the changeable exercise information when clicked
<b>Type:</b>	Manual
<b>Initial State:</b>	Each exercise in the workout section is displayed with an edit exercise button
<b>Input:</b>	click edit exercise button
<b>Output:</b>	A fillable dialog box is launched with information of the exercise. Once the edit exercise button is clicked the dialog box will close and update the exercise information in the list of exercises for the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #31:</b>	<b>FR-WS-5</b>
<b>Description:</b>	Testing that the Workout section prompts the user to add repetitions and sets of each exercise logged in the current calendar day
<b>Type:</b>	Manual
<b>Initial State:</b>	Workout section is displayed with list of exercises for current calendar day
<b>Input:</b>	An event that loads the workout section
<b>Output:</b>	If repetition and sets for exercises not logged then dialog box for exercise is launched and the missing repetition and set values are highlighted
<b>Expected:</b>	
<b>Result:</b>	PASS

### 3.6 Rest Section Page

<b>Test #32:</b>	<b>FR-RS-1</b>
<b>Description:</b>	Testing that Rest section launches with sleep statistics of current calendar day
<b>Type:</b>	Manual
<b>Initial State:</b>	Rest section is initialized with the sleep statistics of the current calendar day
<b>Input:</b>	An event that loads the rest section
<b>Output:</b>	Sleep statistics are loaded for the current calendar day
<b>Expected:</b>	
<b>Result:</b>	PASS

<b>Test #33:</b>	<b>FR-RS-2</b>
<b>Description:</b>	Testing that user can alter inaccurate sleep data
<b>Type:</b>	Manual
<b>Initial State:</b>	Rest section is initialized with the sleep statistics of the current calendar day
<b>Input:</b>	Alter sleep data
<b>Output:</b>	The sleep data is updated with user changes
<b>Expected:</b>	
<b>Result:</b>	PASS

## 4 Nonfunctional Requirements Evaluation

### 4.1 Usability

### 4.2 Performance

### 4.3 etc.

## 5 Comparison to Existing Implementation

This section will not be appropriate for every project.

## 6 Unit Testing

### 6.1 Workout Section

Unit tests for the workout section: <https://github.com/BillNguyen1999/REVITALIZE/blob/main/src/SERVER/backend/test/exercise.test.js>.

Test ID	FR	Inputs	Expected Values	Actual Values	Result
WS1	FR-WS-1 and FR-WS-5	{email: 'test@gmail.com', dateAdded: '2022-01-01'}	[ name: 'Exercise 1', name: 'Exercise 2' ]	[ name: 'Exercise 1', name: 'Exercise 2' ]	PASS
WS2	FR-WS-1 and FR-WS-5	{email: 'fail@gmail.com', dateAdded: '2022-01-01'}	'Error in getting exercise list'	'Error in getting exercise list'	PASS
WS3	FR-WS-2	{ success: true, message: 'Success in adding exercise data', id: 'exerciseid', email: 'test@gmail.com', name: 'Test Exercise', sets: 3, repetitions: 10, weight: 50, dateAdded: '2022-03-07' }			PASS
WS4	FR-WS-3	{email: 'test@gmail.com', dateAdded: '2022-01-01', name: 'push-ups'}	{success: true, message: 'Success in deleting exercise data'}	{success: true, message: 'Success in deleting exercise data'}	PASS
WS5	FR-WS-3	{email: 'not-found@gmail.com', dateAdded: '2022-01-01', name: 'push-ups'}	{success: false, message: 'Was not able to delete selected exercise data'}	{success: false, message: 'Was not able to delete selected exercise data'}	PASS

Table 1: Workout Section Unit Tests Part 1

Test ID	FR	Inputs	Expected Values	Actual Values	Result
WS6	FR-WS-4	params: { email: 'example@gmail.com', dateAdded: '2022-01-01', name: 'exercise-Name' }, body: { reps: 10, sets: 3 }	{success: true, message: 'Success in editing exercise data'}	{success: true, message: 'Success in editing exercise data'}	PASS
WS7	FR-WS-4	params: { email: 'not-found@gmail.com', dateAdded: '2022-03-07', name: 'push-ups' }, body: { sets: 3, reps: 10 }	{success: false, message: "Was not able to find appropriate exercise data to edit" }	{success: false, message: "Was not able to find appropriate exercise data to edit" }	PASS
WS8	FR-WS-1 and FR-WS-5	{email: test@gmail.com, name: 'pushup', dateAdded: 2022-01-01 }	{success: true, message: 'Success in getting exercise data' }	{success: true, message: 'Success in getting exercise data' }	PASS
WS9	FR-WS-1 and FR-WS-5	{email: 'test@gmail.com', name: 'pushup', dateAdded: 'invalid-date' }	{success: false, message: 'Error in getting exercise data' }	{success: false, message: 'Error in getting exercise data' }	PASS

Table 2: Workout Section Unit Tests Part 2



## 6.2 Rest Section

Unit tests for the rest section: <https://github.com/BillNguyen1999/REVITALIZE/blob/main/src/SERVER/backend/test/sleep.test.js>.

Test ID	FR	Inputs	Expected Values	Actual Values	Result
RS1	FR-RS-1 and FR-RS-2	{email: 'test@gmail.com', dateAdded: '2022-01-01'}	{success: true, message: 'Success in getting sleep data'}	{success: true, message: 'Success in getting sleep data'}	PASS
RS2	FR-RS-1 and FR-RS-2	{email: 'test@gmail.com', dateAdded: 'invalid-date'}	{success: false, message: 'Error in getting sleep data'}	{success: false, message: 'Error in getting sleep data'}	PASS
RS3	FR-RS-1	{ success: true, message: 'Success in adding sleep data', id: 'sleepid', email: 'test@gmail.com', sleepHour: 12, bedHour: 10, sleepMinute: 5, bedMinute: 5, dateAdded: '2022-03-07'}			PASS
RS4	FR-RS-2	{email: 'test@gmail.com', dateAdded: '2022-01-01'}	{success: true, message: 'Success in deleting sleep data'}	{success: true, message: 'Success in deleting sleep data'}	PASS
RS5	FR-RS-2	{email: 'not-found@gmail.com', dateAdded: '2022-01-01'}	{success: false, message: 'Was not able to delete selected sleep data'}	{success: false, message: 'Was not able to delete selected sleep data'}	PASS

Table 3: Rest Section Unit Tests Part 1

Test ID	FR	Inputs	Expected Values	Actual Values	Result
RS6	FR-RS-2	params:{ email: 'example@gmail.com', dateAdded: '2022-01-01'}, body: { sleep-Hour: 12, bed-Hour: 11, sleep-Minute: 57, bedMinute: 47}	{success: true, message: 'Success in editing sleep data'}	{success: true, message: 'Success in editing sleep data'}	PASS
RS7	FR-RS-2	params: { email: 'not-found@gmail.com', dateAdded: '2022-03-07'}, body: { sleep-Hour: 12, bed-Hour: 11, sleep-Minute: 57, bedMinute: 47}	{success: false, message: "Was not able to find appropriate sleep data to edit" }	{success: false, message: "Was not able to find appropriate sleep data to edit" }	PASS

Table 4: Rest Section Unit Tests Part 2

### 6.3 Diet Section

Unit tests for the rest section: <https://github.com/BillNguyen1999/REVITALIZE/blob/main/src/SERVER/backend/test/foodLog.test.js>.

Test ID	FR	Some Inputs	Some Expected Values	Corresponding Actual Values	Result
DS1	FR-DS-3	{email: 'test@gmail.com', foodDate: 2022-03-08}	{success: true, message: 'Success in getting food log'}	{success: true, message: 'Success in getting food log'}	PASS
DS2	FR-DS-2	{email: 'test@gmail.com', foodDate: 2022-03-08, calories: 1}	{success: true, message: 'Meal successfully added', calories: 1}	{success: true, message: 'Meal successfully added', calories: 1}	PASS
DS3	FR-DS-3	{email: 'test@gmail.com', foodDate: 2022-03-08, foodName: 'name'}	{success: true, message: 'Success in deleting meal'}	{success: true, message: 'Success in deleting meal'}	PASS
DS4	FR-RS-8	{email: 'test@gmail.com', foodDate: 2022-03-08}	{success: true, message: 'Success in updating meal'}	{success: true, message: 'Success in updating meal'}	PASS

Table 5: Diet Section Unit Tests Part 1

## 7 Changes Due to Testing

Formal testing did not reveal any necessary changes in terms of module interfacing, decomposition, or internal design. Changes made to code were to address bugs and logical errors revealed by the testing plan. User interface improvements were made throughout the development process in response to feedback from developers and informal testers.

## 8 Automated Testing

## 9 Trace to Requirements

Table 6: **Traceability Matrix for Login Page Functional Requirements**

		Requirements								
		FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9
Test Cases	FR-LP-1	X								
	FR-LP-2		X							
	FR-LP-3			X						
	FR-LP-4				X					
	FR-LP-5					X				
	FR-LP-6						X			
	FR-LP-7							X		
	FR-LP-8								X	
	FR-LP-9									X

Table 7: **Traceability Matrix for Signup Page Functional Requirements**

		Requirements				
		FR10	FR11	FR12	FR13	FR14
Test Cases	FR-SP-1	X				
	FR-SP-2		X			
	FR-SP-3			X		
	FR-SP-4				X	
	FR-SP-5					X

Table 8: **Traceability Matrix for Main Page Functional Requirements**

		Requirements				
		FR15	FR16	FR17	FR18	FR30
Test Cases	FR-MP-1	X				
	FR-MP-2		X			X
	FR-MP-3			X		
	FR-MP-4				X	

Table 9: **Traceability Matrix for Diet Page Functional Requirements**

		Requirements								
		FR19	FR20	FR21	FR22	FR23-25	FR26	FR27	FR28	FR29
Test Cases	FR-DS-1	X								
	FR-DS-2		X							
	FR-DS-3			X						
	FR-DS-4				X					
	FR-DS-5					X				
	FR-DS-6						X			
	FR-DS-7							X		
	FR-DS-8								X	X

Table 10: **Traceability Matrix for Workout Page Functional Requirements**

		Requirements				
		FR31	FR32	FR33	FR34	FR35
Test Cases	FR-WP-1	X				
	FR-WP-2		X			
	FR-WP-3			X		
	FR-WP-4				X	
	FR-WP-5					X

Table 11: **Traceability Matrix for Rest Section Functional Requirements**

		Requirements	
		FR36	FR37
Test Cases	FR-RS-1	X	
	FR-RS-2		X

Table 12: **Traceability Matrix for Look and Feel Nonfunctional Requirements**

		Requirements	
		LF1	LF2
Test Cases	NFR-LF1	X	
	NFR-LF22		X

Table 13: **Traceability Matrix for Usability and Humanity Nonfunctional Requirements**

		Requirements					
		UH1	UH2	UH3	UH4	UH5	UH6
Test Cases	NFR-UH1	X					
	NFR-UH2		X				
	NFR-UH3			X			
	NFR-UH4				X		
	NFR-UH5					X	

Table 14: **Traceability Matrix for Performance Nonfunctional Requirements**

		Requirements				
		PE1	PE2	PE3	PE4	PE5
Test Cases	NFR-PE1	X				
	NFR-PE2		X			
	NFR-PE3				X	
	NFR-PE4					X

Table 15: **Traceability Matrix for Operational Nonfunctional Requirements**

		Requirements	
		OE1	OE2
Test Cases	NFR-OE1	X	

Table 16: **Traceability Matrix for Maintainability and Portability Nonfunctional Requirements**

		Requirements		
		MP1	MP2	MP3
Test Cases	NFR-MP1	X		
	NFR-MP2		X	

Table 17: **Traceability Matrix for Security Nonfunctional Requirements**

		Requirements	
		SE1	SE2
Test Cases	NFR-SE1	X	
	NFR-SE2		X

Table 18: **Traceability Matrix for Cultural and Political Nonfunctional Requirements**

		Requirements
		CU1
Test Cases	NFR-CU1	X



## 10 Trace to Modules

Req.	Modules
FR-LP-1	M3
FR-LP-2	M3
FR-LP-3	M3
FR-LP-4	M3
FR-LP-5	M3
FR-LP-6	M3
FR-LP-7	M3
FR-LP-8	M3
FR-LP-9	M3
FR-SP-1	M18
FR-SP-2	M18
FR-SP-3	M18
FR-SP-4	M18
FR-SP-5	M18
FR-MP-1	M1
FR-MP-2	M1
FR-MP-3	M1
FR-MP-4	M1
FR-DS-1	M7
FR-DS-2	M7
FR-DS-3	M7
FR-DS-4	M8
FR-DS-5	M8, M10
FR-DS-6	M11
FR-DS-7	M11
FR-DS-8	M9
FR-WP-1	M14
FR-WP-2	M14
FR-WP-3	M15
FR-WP-4	M15
FR-WP-5	M17
FR-RS-1	M5
FR-RS-2	M6

Table 19: Trace Between Requirements and Modules

## 11 Code Coverage Metrics

## 12 Reflection Appendix

Bill Nguyen: for the vnv plan, it was more formulation rather than implementation, we looked at how we were going to test our project rather than actually doing it. For the vnv report it was more the implementation of our formulation where we wrote actual unit/automated tests and tested our project fully and then compared it to our vnv plan to see what requirements etc. did we satisfy and maybe find things we need to improve on.

Hasan Kibria: In comparison to the vnv plan, the vnv report was more based on practicality an implementation. To complete it fully, there was real code and test cases that had to be thought of an implemented so that they could then be documented in the vnv report. In the vnv plan it was more of an outlook of what we envisioned our testing to look like.

Syed Bokhari: The VNV plan focuses on formulating the testing approach and strategies, while the VNV report is more concerned with the implementation and documentation of the actual testing process. The VNV report involves the creation and execution of test cases, which are then compared to the plan to identify any gaps or areas for improvement. The VNV plan provides a high-level view of the testing process, while the VNV report is a more detailed account of the actual testing activities.

Youssef Dahab: Both the VnV plan and VnV report take inspiration from the functional and non-functional requirements in the SRS document. The VnV plan described how we were going to test our functional and non-functional requirements while the VnV report described the results of performing those tests.