

# Verification and Validation Report: REVITALIZE

Author Name

March 6, 2023

# 1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

## 2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

# 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>
<b>7</b>	<b>Changes Due to Testing</b>	<b>17</b>
<b>8</b>	<b>Automated Testing</b>	<b>18</b>
<b>9</b>	<b>Trace to Requirements</b>	<b>18</b>
<b>10</b>	<b>Trace to Modules</b>	<b>18</b>
<b>11</b>	<b>Code Coverage Metrics</b>	<b>18</b>

## List of Tables

## List of Figures

This document ...

## 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
<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

## 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
- 10 Trace to Modules
- 11 Code Coverage Metrics