Project Title: System Verification and Validation Plan for REVITALIZE

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

November 2, 2022

1 Revision History

Date	Version	Notes
October 31st, 2022	Bill Nguyen	Added Functional Requirements Tests for Login Page
October 31st, 2022	Bill Nguyen	Added Non Functional Requirements Tests for Look and Feel, Usability and Performance
October 31st, 2022	Bill Nguyen	Added 3 Questions to Usability Survey
November 1st, 2022	Youssef Dahab	Added VnV Team, SRS Verification Plan
November 1st, 2022	Bill Nguyen	Self Reflection
November 1st, 2022	Youssef Dahab	Added Design Verification Plan

Contents

List of Tables

[Remove this section if it isn't needed —SS]

List of Figures

[Remove this section if it isn't needed —SS]

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can simply reference the SRS (**?**) tables, if appropriate —SS]

This document ... [provide an introductory blurb and roadmap of the Verification and Validation plan —SS]

3 General Information

3.1 Summary

[Say what software is being tested. Give its name and a brief overview of its general functions. —SS]

3.2 Objectives

[State what is intended to be accomplished. The objective will be around the qualities that are most important for your project. You might have something like: "build confidence in the software correctness," "demonstrate adequate usability." etc. You won't list all of the qualities, just those that are most important. —SS]

3.3 Relevant Documentation

[Reference relevant documentation. This will definitely include your SRS and your other project documents (MG, MIS, etc). You can include these even before they are written, since by the time the project is done, they will be written. —SS]

?

4 Plan

[Introduce this section. You can provide a roadmap of the sections to come. —SS]

4.1 Verification and Validation Team

[You, your classmates and the course instructor. Maybe your supervisor. You shoud do more than list names. You should say what each person's role is for the project. A table is a good way to summarize this information. —SS]

Name	Role	Description
Syed Bokhari	Front-end Application Tester	Run tests to verify and validate front-end design of the application.
Bill Nguyen	Back-end Application Tester	Run tests to verify and validate back-end design of the application.
Hasan Kibria	Database Application Tester	Run tests to verify and validate database schema, tables, relationships, and data mappings.
Logan Brown	Performance Tester	Run tests to verify and validate application speed, responsiveness, and stability.
Youssef Dahab	UI/UX Tester	Run tests to verify and validate application's accessibility, usability, efficiency, and safety to measure HCI aspects.
Mahmoud Anklis	System Integration Tester	Run tests to verify and validate front-end, back-end, and database components integration with each other.
Primary & Secondary Stakeholders	UAT	Perform user acceptance testing to verify and validate that the application meets its purpose.
Dr. Smith	Course Instructor	Provide high level project feedback.
Ting-Yu Wu	Course TA	Provide low level project feedback.

REVITALIZE team members are aware that they might have to put on multiple hats or perform multiple roles.

4.2 SRS Verification Plan

[List any approaches you intend to use for SRS verification. This may just be ad hoc feedback from reviewers, like your classmates, or you may have something more rigorous/systematic in mind.. —SS]

[Remember you have an SRS checklist —SS]

The REVITALIZE team's SRS verification plan is to have the SRS reviewed by classmates, group members, and course TA. Classmates already

adhocly reviewed REVITALIZE's SRS and provided their feedback. Furthermore, the course TA will review REVITALIZE's SRS to provide more detailed and structured feedback. REVITALIZE team members will meet together to gather all feedback from classmates, other team members and course TA. The team will then assess the feedback and analyze what can be incorporated. After that, the SRS will be refined and updated as part of the SRS verification process.

4.3 Design Verification Plan

[Plans for design verification —SS]

[The review will include reviews by your classmates —SS]

[Remember you have MG and MIS checklists —SS]

The REVITALIZE team's design verification plan is to have its design reviewed by classmates, group members, and course TA. The team will gather all feedback from classmates, other group members, and the course TA and incorporate it in the design document. After completing the MG and MIS, the team will ensure that their modules match the system requirements in the SRS. Finally, a design walk-through will be conducted by the team to verify the design.

4.4 Verification and Validation Plan Verification Plan

[The verification and validation plan is an artifact that should also be verified. —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

4.5 Implementation Verification Plan

[You should at least point to the tests listed in this document and the unit testing plan. —SS]

[In this section you would also give any details of any plans for static verification of the implementation. Potential techniques include code walk-throughs, code inspection, static analyzers, etc. —SS]

4.6 Automated Testing and Verification Tools

[What tools are you using for automated testing. Likely a unit testing framework and maybe a profiling tool, like ValGrind. Other possible tools include a static analyzer, make, continuous integration tools, test coverage tools, etc. Explain your plans for summarizing code coverage metrics. Linters are another important class of tools. For the programming language you select, you should look at the available linters. There may also be tools that verify that coding standards have been respected, like flake9 for Python. —SS]

[The details of this section will likely evolve as you get closer to the implementation. —SS]

4.7 Software Validation Plan

[If there is any external data that can be used for validation, you should point to it here. If there are no plans for validation, you should state that here. —SS]

5 System Test Description

5.1 Tests for Functional Requirements

Subsections of the requirements will be divided into the events from our SRS, which are Login Page, Sign up Page, Main Page, Diet Section, Workout Section and Rest Section. There will be 1 test per functional requirement, and will follow the same order as functional requirements in SRS (ex. FR1 in VnV plan is the test for FR1 in SRS).

5.1.1 Login Page Testing

Testing all functional requirements for login page of REVITALIZE. (Refer to BE1 in SRS)

1. FR-LP-1

   Control: Manual

   Initial State: Loading stage of the login page

Input: An event that loads the login page

Output: Login page is displayed with all necessary components

Test Case Derivation: Request is made to load login page

How test will be performed: Tester will open REVITALIZE application and login page should be displayed

2. FR-LP-2

   Control: Manual

   Initial State: Login page is displayed with username textbox

   Input: Enter username information in textbox

   Output: Username information entered is displayed in textbox

   Test Case Derivation: User can enter information in username textbox

   How test will be performed: Tester will enter information in username textbox and checks if textbox displays what the tester entered.

3. FR-LP-3

   Control: Manual

   Initial State: Login page is displayed with password textbox

   Input: Enter password information in textbox

   Output: password information entered is displayed in textbox via hidden text

   Test Case Derivation: User can enter information in password textbox

   How test will be performed: Tester will enter information in password textbox and checks if textbox displays what the tester entered via hidden text.

4. FR-LP-4

   Control: Manual

   Initial State: Login page is displayed with login button

   Input: Click login button

Output: Intended events occurs. Refer to FR9

Test Case Derivation: User clicks login button and a request is made based on username and password text-boxes

How test will be performed: Tester will click on login button and check if request is made correctly

5. FR5

   Control: Manual

   Initial State: Login page is displayed with forgot password button

   Input: Click forgot password button

   Output: Display forgot password screen with textbox to enter email

   Test Case Derivation: User clicks forgot password button and request is made to display forgot password screen with textbox to enter email

   How test will be performed: Tester will click on forgot password button and checks if forgot password screen is displayed with textbox to enter email

6. FR-LP-6

   Control: Manual

   Initial State: Login page is displayed with stay logged in checkbox that is empty

   Input: Click stay logged in checkbox

   Output: 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

   Test Case Derivation: User clicks on stay logged in checkbox and displays appropriate action

   How test will be performed: Tester will click on checkbox and checks to see if check-mark is displayed if checkbox was empty and if an empty checkbox appears if checkbox contained a check-mark

7. FR-LP-7

   Control: Manual

   Initial State: Loading stage of REVITALIZE where previous state had stay logged in checkbox checked

   Input: An event that loads REVITALIZE

   Output: Display main page, with same data from previous state of main page

   Test Case Derivation: User can load REVITALIZE and main page is displayed with same data as the previous time user opened REVITALIZE main page

   How test will be performed: Tester will check stay logged in checkbox go to main page, leave REVITALIZE, reopen REVITALIZE and check whether same data from main page is the same from the last time tester opened main page

8. FR-LP-8

   Control: Manual

   Initial State: Login page is displayed with sign up button

   Input: Click sign up button

   Output: Loads and displays sign up page

   Test Case Derivation: User can click sign up button which loads and displays sign up page

   How test will be performed: Tester will click on sign up button and checks if sign up page is displayed

9. FR-LP-9

   Control: Manual

   Initial State: Login page is displayed with inputted information in username and password text-boxes

   Input: Click login button

Output: if failure state, display an invalid password or username banner, else if success state, load and display main page

Test Case Derivation: User clicks login button and request is made based on username and password, and will proceed to main page only if username and password are valid

How test will be performed: Tester will click on login button and test for scenarios when login should be successful and when login should fail

5.1.2 Signup Page Testing

Testing all functional requirements for sign page of REVITALIZE. (Refer to BE2 in SRS)

1. FR-SP-1

   Control: Manual

   Initial State: Signup page is displayed with username textbox

   Input: Enter username information in textbox

   Output: Username information entered is displayed in textbox

   Test Case Derivation: User can enter information in username textbox

   How test will be performed: Tester will enter information in username textbox and checks if textbox displays what the tester entered

2. FR-SP-2

   Control: Manual

   Initial State: Signup page is displayed with password textbox

   Input: Enter password information in textbox

   Output: Password information entered is displayed in textbox via hidden text

   Test Case Derivation: User can enter information in password textbox

   How test will be performed: Tester will enter information in password textbox and checks if textbox displays what the tester entered via hidden text

3. FR-SP-3

   Control: Manual

   Initial State: signup page is displayed with email textbox

   Input: Enter email information in textbox

   Output: Email information entered is displayed in textbox

   Test Case Derivation: User can enter information in email textbox

   How test will be performed: Tester will enter information in email textbox and checks if textbox displays what the tester entered

4. FR-SP-4

   Control: Manual

   Initial State: Signup page is displayed with signup button

   Input: Click signup button

   Output: Intended events occurs. Refer to FR14

   Test Case Derivation: User clicks signup button and a request is made based on username, password and email text-boxes

   How test will be performed: Tester will click on signup button and check if request is made correctly

5. FR-SP-5

   Control: Manual

   Initial State: Signup page is displayed with inputted information in username and password text-boxes

   Input: Click signup button

   Output: if failure state, display an invalid username, password or email banner, else if success state, load and display login page

   Test Case Derivation: User clicks signup button and request is made based on username, password and email, and will proceed to login page only if username, password and email are valid

How test will be performed: Tester will click on signup button and test for scenarios when signup should be successful and when signup should fail

5.1.3 Main Page Testing

Testing all functional requirements for main page of REVITALIZE. (Refer to BE3 in SRS)

1. FR-MP-1

   Control: Manual

   Initial State: Main page is displayed with calender of current date

   Input: An event that loads the main page

   Output: Main page is displayed with all necessary components

   Test Case Derivation: Request is made to load main page

   How test will be performed: Tester will successfully login to the REVITALIZE application and will visually check if the correct calender data is loaded

2. FR-MP-2

   Control: Manual

   Initial State: Main page and Diet, Workout, Rest sections are displayed with previous day and next day buttons

   Input: An event that loads the main page, Diet, Workout, Rest sections and the previous day and next day buttons are clicked

   Output: Main page, Diet, Workout, Rest sections are displayed with previous day and next day buttons. Once the next day button is clicked, the calender refreshes the calender information for the next day. Once the preivous day button is clicked, the calender refreshes the calender information for the previous day.

   Test Case Derivation: Request is made to load main page, Diet, Workout, Rest sections and the previous day and next day buttons are clicked

How test will be performed: Tester will successfully login to the REVITALIZE application and will visually check if the previous day and next day buttons are visible. The tester will click the previous day button and visually check if the calender is updated to the date of the previous day. The tester will click the next day button and visually check if the calender is updated to the date of the next day. The tester will enter the Diet, Workout and Rest sections and will visually check if the previous day and next day buttons are visible. The tester will click the previous day button and visually check if the calender is updated to the date of the previous day. The tester will click the next day button and visually check if the calender is updated to the date of the next day

3. FR-MP-3

   Control: Manual

   Initial State: Each interaction after leaving the main page must have a visible back button

   Input: An event that loads the next user interface after leaving the main page and the back button is clicked

   Output: The next user interface after leaving the main page iis displayed with a back button. Once the back button is clicked, the main page is loaded

   Test Case Derivation: Request is to leave the main page and the back button is clicked

   How test will be performed: Tester will leave the main page by selecting any of the options on the page. The tester will visually check if a back button is visible on every page that is entered through the main page interaction. The tester will click the back button and visually check if the current page is closed and the main page is loaded. The tester will repeat this process with every page that is loaded from clicking an interaction from the main page

4. FR-MP-4

   Control: Manual

Initial State: Main page is displayed with Diet, Exercise and Rest buttons available to click

Input: An event that loads the main page and the Diet, Exercise and Rest buttons are clicked

Output: 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

Test Case Derivation: Request is made to load main page and the Diet, Exercise and Rest buttons are clicked

How test will be performed: Tester will successfully login to the REVITALIZE application and will visually check if the Diet, Exercise and Rest buttons are visible. The tester will click the Diet button and visually check if the Diet interface is loaded. the tester will click the Exercise button and visually check if the Exercise interface is loaded. The tester will click the Rest button and visually check if the Rest interface is loaded

5.1.4 Diet Section Testing

Testing all functional requirements for rest section of REVITALIZE. (Refer to BE4 in SRS)

1. FR-DS-1

   Control: Manual

   Initial State: Diet section is initialized for the first time and an initial information dialog is launched

   Input: An event that loads the diet section for the first time

   Output: A fillable dialog box is launched with height, dietary information, weight and calorie information

   Test Case Derivation: Request is made to enter diet section for the first time

   How test will be performed: Tester will enter rest section for the first time and visually check if the dialog box with correct input information is launched

2. FR-DS-2

   Control: Manual

   Initial State: Diet section is initialized for the first time and an initial information dialog is launched

   Input: Initial information dialog values are filled

   Output: Initial information values are saved to the database

   Test Case Derivation: Initial information dialog is filled

   How test will be performed: Tester will fill the initial information dialog after entering the diet section. The tester will visually check the user data in the database to see if the initial information is saved

3. FR-DS-3

   Control: Manual

   Initial State: section section is initialized with a list of food logged for the current calender day

   Input: An event that loads the rest section

   Output: A list of inputted food is loaded for the current calender day

   Test Case Derivation: Request is made to enter rest section

   How test will be performed: Tester will enter the rest section and will visually check if a list of logged food is loaded for the current calender day

4. FR-DS-4

   Control: Manual

   Initial State: Diet section is displayed with add food button

   Input: Click add food button

   Output: A user interface is launched that lets the user select between searching for food or adding a custom meal

   Test Case Derivation: User clicks add food button

How test will be performed: Tester will click on add food button and will visuallly check if the user interface with options for adding a custom meal and searching for food are available.

5. FR-DS-5

   Control: Manual

   Initial State: Food adding user interface is displayed with search food button

   Input: Click add search food button

   Output: A recipe criteria user interface is launched that displays a list of modifiable criteria and a search button

   Test Case Derivation: User clicks search food button

   How test will be performed: Tester will click on search food exercise button and will visuallly check if a recipe criteria user interface is launched that displays a list of modifiable criteria and a search button

6. FR-DS-6

   Control: Manual

   Initial State: Recipe criteria user interface is launched

   Input: Search criteria is modified and search button is clicked

   Output: List of recipes are loaded correctly based on constraints of search criteria

   Test Case Derivation: User modifies the search criteria and clicks the search button

   How test will be performed: Tester will modify the search criteria and click the search button. The tester will visually check if a recipe list is loaded. The tester will visually check the correcctness of the list based on selected constraints.

7. FR-DS-7

Control: Manual

Initial State: Recipe list is loaded based on search constraints

Input: Add recipe button is clicked

Output: Selected recipe is added to the list of food logged on the current calender day

Test Case Derivation: User selects the recipe from the recipe list and clicks the add recipe button

How test will be performed: Tester will select a recipe and click the add recipe button. The tester will visually check if the selected recipe has been added to the list of food logged on the current calender day

8. FR-DS-8

   Control: Manual

   Initial State: Food adding interface is displayed with add custom meal button

   Input: Click add custom meal button

   Output: A dialog box is launched that lets the user fill custom meal information. the meal is added to the food log list of the current calender day

   Test Case Derivation: User clicks add custom meal button

   How test will be performed: Tester will click on add custom meal button and will visuallly check if a fillable dialog box is launched. The tester will fill the dialog box information and click the add custom meal button. The tester will visually check if the meal has been added to the list of logged food for the current calender day

5.1.5 Workout Section Testing

Testing all functional requirements for workout section of REVITALIZE. (Refer to BE5 in SRS)

1. FR-WS-1

   Control: Manual

Initial State: Workout section is initialized with a preset list of exercises of the current calender day

Input: An event that loads the workout section

Output: A preset list of exercises is loaded for the current calender day

Test Case Derivation: Request is made to enter workout section

How test will be performed: Tester will enter the workout section and will visually check if a list of preset exercises are loaded for the current calender day

2. FR-WS-2

   Control: Manual

   Initial State: Workout section is displayed with add exercise button

   Input: Click add exercise button

   Output: A dialog box is launched that lets the user fill custom exercise information. the exercise is added to the exercise list of the current calender day

   Test Case Derivation: User clicks add exercise button

   How test will be performed: Tester will click on add exercise button and will visuallly check if a fillable dialog box is launched. The tester will fill the dialog box information and click the add exercise button. The tester will visually check if the exercise has been added to the list of exercises for the current calender day

3. FR-WS-3

   Initial State: Each exercise in the workout section is displayed with a delete exercise button

   Input: Click delete exercise button

   Output: The exercise is deleted from the exercise list of the current calender day

   Test Case Derivation: User clicks delete exercise button

How test will be performed: Tester will click on delete exercise button and will visuallly check if the exercise is deleted from the list of exercises for the current calender day.

4. FR-WS-4

   Initial State: Each exercise in the workout section is displayed with an edit exercise button

   Input: click edit exercise button

   Output: 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 calender day

   Test Case Derivation: User clicks edit exercise button

   How test will be performed: Tester will click on edit exercise button and will change the infromation on the fillable dialog box. The tester will click the edit exercise button and will visually check if the information is changed on the exercise list for the current calender day.

5. FR-WS-5

   Control: Manual

   Initial State: Workout section is displayed with list of exercises for current calender day

   Input: An event that loads the workout section

   Output: If repitition and sets for exercises not logged, dialog box for exercise is launched and the missing repitition and set values are highlighted

   Test Case Derivation: User launches the workout section. User adds an exercise and does not input repitition and set values.

   How test will be performed: Tester will add exercise without inputting values for set and repititions. The user will visually check if the exercise dialog is launched with highlighted missing repititions and set values.

5.1.6 Rest Section Testing

Testing all functional requirements for rest section of REVITALIZE. (Refer to BE6 in SRS)

1. FR-RS-1

   Control: Manual

   Initial State: Rest section is initialized with the sleep statistics of the current calender day

   Input: An event that loads the rest section

   Output: Sleep statistics are loaded for the current calender day

   Test Case Derivation: Request is made to enter rest section

   How test will be performed: Tester will enter the rest section and will visually check if slep statistics are loaded for the current calender day

2. FR-RS-2

   Control: Manual

   Initial State: Rest section is initialized with the sleep statistics of the current calender day

   Input: Alter sleep data

   Output: The sleep data is updated with user changes

   Test Case Derivation: User alters sleep data

   How test will be performed: Tester will alter the sleep data and will visually check if the new values are correctly displayed in the sleep statistics

...

5.2 Tests for Nonfunctional Requirements

[The nonfunctional requirements for accuracy will likely just reference the appropriate functional tests from above. The test cases should mention reporting the relative error for these tests. —SS]

[Tests related to usability could include conducting a usability test and survey. —SS]

5.2.1 Look and Feel Testing

1. LF1

   Type: Dynamic, Functional, Manual

   Initial State: User is using REVITALIZE features

   Input/Condition: REVITALIZE features are in use

   Output/Result: All UI/UX design and elements matches original design and are displayed correctly and neatly

   How test will be performed: All related stakeholders will test application with the focus on neatness and attractiveness of UI/UX design of REVITALIZE and answer question 1 of the Usability Survey. Would need an average rating of 8.5 or above out of 10 and assess all stakeholders responses to make improvements

2. LF2

   Type: Static, Manual

   Initial State: A display of all pages in REVITALIZE

   Input/Condition: All displays for all pages in REVITALIZE are at a common point during a user session

   Output/Result: All colours are considered appealing, contrasting and non-intrusive

   How test will be performed: All related stakeholders will test application with the focus on colour and answer question 2 of the Usability Survey. Would need an average rating of 8.5 or above out of 10 for each factor and assess all stakeholders responses to make improvements

5.2.2 Usability and Humanity Testing

1. UH1

   Type: Dynamic, Functional, Manual

   Initial State: User is using REVITALIZE features

Input/Condition: REVITALIZE features are in use, using one hand/one finger

Output/Result: REVITALIZE features are displaying correct outputs and results

How test will be performed: All related stakeholders with varying size hands/fingers can use all aspects of REVITALIZE using one hand/finger and have an average rating of 9.5 or above for question 3 of the Usability Survey

2. UH2

   Type: Dynamic, Functional, Manual

   Initial State: Main page of application is displayed

   Input/Condition: User uses main page to access features (Diet, Workout and Rest Section)

   Output/Result: All features take less than 10 seconds to access

   How test will be performed: Stakeholders will navigate to the Diet, Workout and/or Rest section from the main page in 10 seconds or less. 90% of stakeholders need to be able to navigate to any of the sections in 10 seconds or less

3. UH3

   Type: Dynamic, Functional, Manual

   Initial State: REVITALIZE is loaded but not in use

   Input/Condition: Users in targeted demographic will use all features of REVITALIZE

   Output/Result: Results gathered from survey

   How test will be performed: Primary stakeholders such as teenagers and young adults 14 years or older will test application and fill in survey, with the goal of an approval rating of 85% or above

4. UH4

Type: Dynamic, Functional, Manual

Initial State: REVITALIZE is loaded but not in use

Input/Condition: User will use all features of REVITALIZE

Output/Result: User can use and understand basic/common aspects of all features after 3rd iteration

How test will be performed: Stakeholders will use all features/aspects of REVITALIZE and 85% of stakeholders should be able to use and understand basic/common aspects of all features in 3 iterations or less.

5. UH5

Type: Dynamic, Functional, Manual

Initial State: REVITALIZE is loaded but not in use

Input/Condition: User will use all features of REVITALIZE

Output/Result: Results gathered from survey based on consistency of UI aspects such as buttons, drop-downs, words etc.

How test will be performed: Stakeholders will test application with focus on consistency of UI aspects and fill in survey, with the goal of an approval rating of 85% or above

5.2.3 Performance Testing

1. PE1

Type: Dynamic, Functional, Manual

Initial State: REVITALIZE is loaded but not in use

Input/Condition: All REVITALIZE features are loaded with appropriate data

Output/Result: loading time of all REVITALIZE features

How test will be performed: Developers will run performance tests and ensure that all output data loads within 5 seconds or less for 95% of all API responses and outputs

2. PE2

   Type: Dynamic, Functional, Automatic

   Initial State: REVITALIZE is loaded but not in use

   Input/Condition: User uses all features of REVITALIZE where output contain data/numbers

   Output/Result: data/numbers of used features in floating points

   How test will be performed: Developers will run accuracy tests to ensure output data/numbers are accurate for double precision floating points and pass all test cases

3. PE3

   Type: Dynamic, Functional, Manual

   Initial State: REVITALIZE is loaded but not in use

   Input/Condition: Multiple (More than 50) users using REVITALIZE

   Output/Result: Performance metrics (ex. loading time, frames per second etc.)

   How test will be performed: 50 or more users (does not have to be actual people) use/send requests to REVITALIZE application simultaneously and analyze performance trends based on the number of users

4. PE4

   Type: Dynamic, Functional, Manual

   Initial State: REVITALIZE is loaded but not in use

   Input/Condition: User will use all features of REVITALIZE

   Output/Result: Storage percentage of data

   How test will be performed: Developers will try to add as much data as possible to user account and application should be able to store 1 million or more data points for all users

5.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

6 Unit Test Description

[Reference your MIS and explain your overall philosophy for test case selection. —SS] [This section should not be filled in until after the MIS has been completed. —SS]

6.1 Unit Testing Scope

[What modules are outside of the scope. If there are modules that are developed by someone else, then you would say here if you aren't planning on verifying them. There may also be modules that are part of your software, but have a lower priority for verification than others. If this is the case, explain your rationale for the ranking of module importance. —SS]

6.2 Tests for Functional Requirements

[Most of the verification will be through automated unit testing. If appropriate specific modules can be verified by a non-testing based technique. That can also be documented in this section. —SS]

6.2.1 Module 1

[Include a blurb here to explain why the subsections below cover the module. References to the MIS would be good. You will want tests from a black box perspective and from a white box perspective. Explain to the reader how the tests were selected. —SS]

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

2. test-id2

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input:

   Output: [The expected result for the given inputs —SS]

   Test Case Derivation: [Justify the expected value given in the Output field —SS]

   How test will be performed:

3. ...

6.2.2 Module 2

...

6.3 Tests for Nonfunctional Requirements

[If there is a module that needs to be independently assessed for performance, those test cases can go here. In some projects, planning for nonfunctional tests of units will not be that relevant. —SS]

[These tests may involve collecting performance data from previously mentioned functional tests. —SS]

6.3.1 Module ?

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

6.3.2 Module ?

...

6.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

[This is a section that would be appropriate for some projects. —SS]

1. How would you rate overall neatness of UI/UX design of REVITALIZE out of 10? What are elements you like related to neatness? What are elements you would improve related to neatness?

2. Are colours used in REVITALIZE non-intrusive, appealing and/or contrasting? Rate each factor of non-intrusive, appealing and contrasting out of 10? What are elements you like that elevate these factors? What are elements you would improve to elevate these factors?

3. How would you rate the overall ability to use REVITALIZE with only one hand/one finger out of 10? What are elements that help with this? What are elements you would improve?

8 Reflection Appendix

8.1 Knowledge and Skills Needed

Bill Nguyen: My primary role for VnV plan is to be a back-end developer tester, so some skills needed would how to write proper and effective unit tests, for back-end implementation of project, integration tests that tests the project end to end and with the front-end, and also maybe some tests for performance such as load, stress test etc. for back-end code.

8.2 How Knowledge and Skills Will Be Acquired

Bill Nguyen: Knowledge and skills will be acquired by researching techniques on how to write effective unit tests for the back-end for example using the "arrange act assert" and potentially setting minimum thresholds for code/branch coverage to improve consistency and organization of tests. For integration and performance tests also perform research to find effective ways to test back-end with all aspects of project and applying it often.