Software Test Document
for
Vitamin Personalizer

By: Nhat-Huy Tran, Jonathan Yunes, and Ahmet Ege Aytac

Wellness Wizards

July 12, 2025

1. Test Plan: Vitamin Personalizer

1.1 Introduction:

1.1.1 Purpose:

- The purpose of the Software Test Document (STD) is to provide a testing overview for the Vitamin Personalizer. This test plan is designed to make sure the vitamin results are as accurate as possible by testing high, low, and normal inputs. Plus, the software should find lab results that are too low or too high and help fix them so people can be healthier. Many health problems occur because the body lacks sufficient vitamins or nutrients. This software helps set the right vitamin levels for each person based on their weight, height, age, and gender to make them healthier.

1.1.2 Scope:

- The Software Test Document (STD) should give an overview of the features and components that will be tested, such as the user account creation, menu interface, blood work results inputs, and the results given. Additionally, the document will go over various case studies to test components in specific conditions.

1.1.3 Objectives:

- The objective of this document is to verify the performance of the software and make sure that no errors or false recommendations appear. The testing will check all 50 lab tests, make sure the software works correctly, confirm it can tell if results are low, normal, or high, and verify it gives the right health advice for each person based on their age, gender, and body size.

1.2 Test Items

1.2.1 User Account

- Creation and Login of an Account
- Age, Sex, Height, and Weight input checks
- ID verification
- New blood work result input
- Updating the blood work result input

1.2.2 User Interface/Menu Navigation

- Menu navigation

1.2.3 Vitamin Inputs

- There are 50 lists of vitamins and other types of lab tests that are used in the software's database to be input. They are:
  - **WBC:** White blood cells for the immune system
  - **RBC:** Red blood cells for oxygen transport
  - **Hemoglobin:** Oxygen-carrying protein
  - **Hematocrit:** Blood cell percentage
  - **Platelet Count:** Blood clotting
  - **MCV:** Red cell size
  - **MCH:** Hemoglobin per cell
  - **MCHC:** Hemoglobin concentration
  - **RDW:** Red cell size variation

- **MPV:** Platelet size
- **Neutrophils:** Fight bacteria
- **Lymphocytes:** Fight viruses
- **Monocytes:** Remove dead cells
- **Eosinophils:** Fight parasites/allergies
- **Basophils:** Allergic reactions
- **Glucose:** Blood sugar
- **Sodium:** Fluid balance
- **Potassium:** Heart/muscle function
- **Chloride:** Acid-base balance
- **CO2:** Breathing/pH balance
- **BUN:** Kidney function
- **Creatinine:** Kidney function
- **Total Protein:** Liver production
- **Albumin:** Liver synthesis
- **Total Bilirubin:** Liver processing
- **Alkaline Phosphatase:** Liver/bone health
- **ALT:** Liver damage
- **AST:** Liver/heart damage
- **Calcium:** Bone/muscle health
- **Magnesium:** Muscle/nerve function
- **Phosphorus:** Bone/energy metabolism
- **Total Cholesterol:** Heart disease risk
- **Triglycerides:** Fat metabolism
- **HDL Cholesterol:** Good cholesterol
- **LDL Cholesterol:** Bad cholesterol
- **TSH:** Thyroid control hormone
- **Free T4:** Thyroid hormone
- **Free T3:** Active thyroid hormone
- **Vitamin D:** Bone/immune health
- **Vitamin B12:** Nerve/blood health
- **Folate:** DNA/blood formation
- **Iron:** Oxygen transport
- **Ferritin:** Iron storage
- **TIBC:** Iron binding capacity
- **Hemoglobin A1C:** Long-term blood sugar
- **Insulin:** Blood sugar control
- **Uric Acid:** Gout/kidney health
- **CRP:** Inflammation marker
- **ESR:** General inflammation
- **PSA:** Prostate health for men only

1.3 Features to Be Tested:

1.3.1 User management:

- The testing will cover comprehensive user management features, including user registration with personal information such as name, age, sex, height, and weight, along with automatic user ID generation, login functionality, session management, and BMI calculation based on physical measurements.

1.3.2 Lab inputs:

- The lab data input features will be tested to ensure proper entry of all 50 test parameters, data validation for numeric inputs, optional field handling for skipped tests, lab result updating capabilities, and secure data storage in the database.

1.3.3 Vitamin Analyzer/Recommender:

- The analysis engine features require testing of the classification system that determines if lab values are Low, Normal, or High, application of gender-specific reference ranges, age-appropriate value assessment, and comprehensive health analysis generation. The recommendation system features will be evaluated for their ability to provide personalized supplement recommendations, custom dietary advice based on lab results, health improvement suggestions, and proper integration with the clinic database for accurate recommendations.

1.3.4 Database management:

- Database management features, including SQLite database operations, user data storage and retrieval, lab result history tracking, and data integrity validation, will undergo thorough testing.

1.3.5 PDF Generation:

- The PDF generation features will be tested to ensure proper formatting of analysis reports, accurate categorization of normal and abnormal results, source citation for reference ranges, personalized health recommendations display, and printable summary generation functionality for clinic use.

1.4 Features Not to Be Tested:

The following features will not be tested as they are not included in the current software implementation.

1.4.1 User Security:

- Security questions for user account recovery and password reset functionality are excluded from testing.

1.4.2 Vitamin Dictionary:

- A vitamin dictionary or educational content library for users to read about different vitamins and lab tests will not be tested, since this feature is not implemented.

1.4.3 External Application:

- Additional excluded features include multi-language support, data export capabilities, third-party device integration, and email notifications, as these are considered for future development phases.

1.5 Test Approach:

- The testing approach will use three levels: unit testing for individual parts like the analysis engine and database functions, integration testing to verify data flow from input to recommendations, and system testing for complete terminal-based functionality through main.py.

1.5.1 Lab input results

- Users enter lab results via terminal prompts with their personal information, including age, weight, gender, height, and name. Users can leave values empty for tests they don't want analyzed, but the system only accepts numeric values for lab results. After registration, users must save their generated ID to sign in again.

1.5.2 Verification

- Testing will use black-box testing for input-output verification and white-box testing for code logic examination. Functional testing will verify all 50 lab tests classification, numeric input validation, empty field handling, and recommendation generation. Performance testing ensures smooth database operations and terminal interface functionality.

1.6 Pass/Fail Criteria:

1.6.1 Pass Criteria:

- The system passes testing when user registration successfully creates a unique ID that works for login, the system accepts numeric inputs for lab tests while allowing empty fields for skipped tests, and lab results correctly show Low, Normal, or High classifications based on the user's gender and age. The system must provide appropriate recommendations for abnormal lab values, properly save and load user information in the database, generate clear PDF reports showing all results, reject non-numeric inputs with proper error messages, and allow successful login with saved ID while displaying past results. Users must enter values correctly to get accurate results and should recheck their inputs to prevent any harmful recommendations from incorrect data.

1.6.2 Fail Criteria:

- The system fails testing if user registration breaks or loses user data, the system crashes when receiving bad inputs or accessing an empty database, lab values are incorrectly classified as Normal/Low/High, inappropriate or missing recommendations are provided for abnormal values, the database fails to save or retrieve information properly, PDF generation fails or produces unreadable output, the system accepts letters instead of numbers for lab test inputs, login fails with correct credentials or displays wrong user data, or incorrect values are set according to clinic results and professional recommendations are not properly implemented.

1.7 Suspension and Resumption Criteria:

1.7.1 Suspension Criteria:

- Testing will stop if the system crashes frequently, loses user data, gives wrong health advice that could be dangerous, or the terminal stops working completely. Testing will also pause if users cannot register, if login fails, or if PDF reports don't work properly.

1.7.2 Resumption Criteria:

- Testing can start again when all major problems are fixed, the system works without crashing, user data is safe, medical professionals check health recommendations, and all basic functions like registration, login, and PDF creation work correctly.

2. Test Design Specification

2.1.1 Test Case 1.1 Account Registration

- Test case ID: VP-01 AR
- Test case description: This test case is about the user registering an account for the Vitamin Personalizer. The user will input their name, age, sex, height, weight, and BMI, once all the values have been checked, the user will be granted a unique ID for whenever they want to log back in.
- Preconditions: The software should already be running, and a menu should be present for the user to choose whether to register or log in, and the SQL database connector should be functional. All python files should be accessed by the main.py component.
- Test steps: Press 1 as an input to register for an account. Afterwards, the software will allow you to enter your name, age, sex, height, weight, and BMI. Once everything has been entered, press Enter.
- Expected results: The user should be granted a unique ID for their account.
- Actual results: The system successfully reads and registers the user's inputs into the database. The system then generates a unique ID for the user not only to differentiate but as a password for the user to log back into the system.
- Postconditions: The user's account is stored in the SQL user database, and a unique ID is given to both the account and future blood work results in the lab result database. The user will then be allowed to enter their lab inputs.

2.1.2 Test Case 2.1 Adding New Blood Work Results

- Test case ID: VP-02 BL
- Test case description: This test case involves the user entering their blood work results into the Vitamin Personalizer to be analyzed for correct recommendations.
- Preconditions: The software should already be running, and a menu should be present for the user to choose whether to register or log in, and the SQL database connector should be functional. All Python files should be accessed by the main.py component. The user's account is already registered, and they already have their blood work results printed out for the system to enter.
- Test steps: After registering, press 1 for "Enter new blood work results." Once entered, enter the numerical value for all 50 vitamins following your blood work results. You have the option to omit any vitamins you don't want to be analyzed. Once filled in, press Enter.
- Expected results: The system should show the user accurate vitamin recommendations based on the values entered in the system. If there are any empty inputs, the vitamin is omitted in the results screen.
- Actual results: The system has successfully shown the results with the correct vitamin recommendations and diet.
- Postconditions: The user's blood work results have been stored in the blood_test_values database alongside the user's ID.

2.1.3 Test Case 3.1 Updating Blood Work Results

- Test case ID: VP-03 UR
- Test case description: This test case is about updating the user's results with new values so that the new ones replace the old and outdated results.
- Preconditions: The software should already be running, and a menu should be present for the user to choose whether to register or log in, and the SQL database connector should be functional. All Python files should be accessed by the main.py component. The user's account is already registered, and they already have their blood work results printed out for the system to enter. The user has already entered their blood work values first time.
- Test steps: After getting recommendations or logging back in, there will be a menu option to update vitamin values. Press the corresponding number with the option and click Enter. Afterwards, enter your new values for all 50 vitamins. Press Enter when finished.
- Expected results: The user should have new vitamin recommendations based on the newly updated values. The user should be taken to the lab results screen.
- Actual results: The new values have successfully been read and analyzed, and the user is shown the new vitamin recommendation results.
- Postconditions: The user's values in the blood_work_values database are updated to reflect the new vitamin recommendations.

2.1.4 Test Case 4.1 High and Low Value Testing

- Test case ID: VP-04 HL
- Test case description: The test case involves inputting predetermined values for each of the 50 vitamins to test that their conditions work as intended to avoid incorrect recommendations.
- Preconditions: The software should already be running, and a menu should be present for the user to choose whether to register or log in, and the SQL database connector should be functional. All Python files should be accessed by the main.py component. The user's account is already registered, and they already have predetermined vitamin values printed out for the system to enter.
- Test steps: Press the number corresponding to create lab values and press Enter. The system should take you to the screen to enter values for the vitamins. Each vitamin should have a value range that is either low, high, or normal. Enter the predetermined values and press Enter.
- Expected results: The system should correctly analyze values that are either too low, too high, or normal and give the correct vitamin recommendations based on the values given.
- Actual results: The system successfully analyzes and returns the correct recommendations and diet based on the values entered.
- Postconditions: The user's blood work results have been stored in the blood_test_values database alongside the user's ID. The user's values will also be used for future blood work value inputs for better accuracy.

2.1.5 Test Case 5.1 Menu Navigation

- Test case ID: VP-05 MN
- Test case description: This test case is about going through every selection in the menu that the Vitamin Personalizer has, such as selecting to log in, register, enter lab values, or see progress checks.
- Preconditions: The software should already be running, and a menu should be present for the user to choose whether to register or log in, and the SQL database connector should be functional. All Python files should be accessed by the main.py component.
- Test steps: A menu will show up on the screen alongside numbers corresponding to the option. Go through each option by pressing the number associated with it and press Enter.

- Expected results: The system should smoothly transition between screens based on the user's option without crashing or showing the wrong screen.
- Actual results: The system successfully transitions from one screen to the next without breaking or showing the wrong screen.
- Postconditions: The user should be able to navigate through the system with ease.

2.1.6 Test Case 6.1 Account Login Verification

- Test case ID: VP-06 LV
- Test case description: The test case involves testing the account verification system for the Vitamin Personalizer. The user is by default locked out of using it until they either have an account registered or are logged in with an ID.
- Preconditions: The software should already be running, and a menu should be present for the user to choose whether to register or log in, and the SQL database connector should be functional. All Python files should be accessed by the main.py component. The user's account is already registered with a unique ID given to them.
- Test steps: On the startup menu, press 2 to log in and hit Enter. The system will ask for a name and an ID as a password. Enter the registered name and ID into the prompt and press Enter.
- Expected results: The user should be able to log back into the Vitamin Personalizer with their registered ID and name and access the main component.
- Actual results: The user has successfully logged back into the Vitamin Personalizer with their name and ID. The user now has access to the system again.
- Postconditions: The user now has access to the Vitamin Personalizer, where they can enter their blood work values, update their values, or check their progress.

2.1.7 Test Case 7.1 PDF Generation

- Test case ID: VP-07 PG
- Test case description: The test case involves utilizing the PDF generator system in the Vitamin Personalizer, where the user has the option to print out their lab results after inputting their values or going to the progress check menu.
- Preconditions: The software should already be running, and a menu should be present for the user to choose whether to register or log in, and the SQL database connector should be functional. All Python files should be accessed by the main.py component. The user's account is already registered, and they have already entered their lab values.
- Test steps: Once you've obtained your lab results and recommendations, a menu will appear asking if you want to print out your results. Press 1 for yes and Enter afterwards. In the case where you've navigated to the progress check screen, do the same steps mentioned above. A print screen will show up with a preview of the pages. Select your printer option and press the print button.
- Expected results: The user should be able to print out their lab results in physical form.
- Actual results: The user has successfully printed out their lab results on paper, stating all the vitamins that have been analyzed alongside a summary for each.
- Postconditions: The results will be printed on paper, and the user will have the option to print again or go back to the menu, where the user can update their values or go back to the main menu.

2.2 Test Procedures:

2.2.1 Account Registration Procedure

- Procedure ID: VP-01 ARP
- Description: This procedure tests user registration for the Vitamin Personalizer
- Test Environment: Windows Operating system, a browser that can execute Python program.
- Steps to Execute:
    1. Press 1 in the prompt to go to the account registration screen.
    2. Input your account information.
    3. Press Enter once everything has been filled out.
- Post-Execution Actions: The user should have their inputs verified and have their account registered. Afterwards, the user should be allowed to input their vitamin values into the blood_work_values database.

2.2.2 Adding New Blood Work Results Procedure

- Procedure ID: VP-02 BLP
- Description: This procedure tests inputting blood work values into the Vitamin Personalizer.
- Test Environment: Windows Operating system, a browser that can execute Python programs.
- Steps to Execute:
    1. Make sure you are either logged into an existing account or have registered one beforehand
    2. Select the number corresponding to the option “create new lab results” and press Enter.
    3. For all 50 vitamins, enter your blood work values.
    4. For vitamins you want to omit, leave blank.
    5. Press Enter once finished.
- Post-Execution Actions: The user’s inputs should be verified first before being sent to the blood_work_values database and analyzed for correct recommendations.

2.2.3 Updating Blood Work Results Procedure

- Procedure ID: VP-03 URP
- Description: The procedure involves changing and replacing old vitamin values with new and updated values.
- Test Environment: Windows Operating system, a browser that can execute Python programs.
- Steps to Execute:
    1. Make sure you are either logged into an existing account or have registered one beforehand
    2. The user should have already input their blood work results prior.
    3. Select the number corresponding to the option “update lab results” and press Enter.
    4. For all 50 vitamins, enter your blood work values.
    5. For vitamins you want to omit, leave blank.
    6. Press Enter once finished.
- Post-Execution Actions: The user’s inputs should be verified first before being sent to the blood_work_values database and analyzed for correct recommendations. Afterwards, the system should delete the outdated vitamin values and replace them with the newly input values.

2.2.4 High and Low Value Testing Procedure

- Procedure ID: VP-04 HLP
- Description: This procedure involves testing the vitamin analyzer by giving predetermined values to check if it recommends accurately.
- Test Environment: Windows Operating system, a browser that can execute Python program.

- Steps to Execute:
    1. Make sure you are either logged into an existing account or have registered one beforehand
    2. The user should have a predetermined list of vitamin values to input
    3. Select the number corresponding to either “create lab results” or “update lab results” and press Enter.
    4. For all 50 vitamins, enter the predetermined blood work values.
    5. Press Enter once finished.
    6. Compare the results given with the range values for each vitamin.
- Post-Execution Actions: The input should correctly analyze and display the results and recommendations. For low values, it should recommend diets and foods; for high values, it should tell the user to avoid certain foods; and for normal values, it should be separate from the recommendations listed below.

2.2.5 Menu Navigation Procedure

- Procedure ID: VP-05 MNP
- Description: The procedure tests the menu navigation and sees if every option in the menu correctly displays the intended screen for the user.
- Test Environment: Windows Operating system, a browser that can execute Python programs.
- Steps to Execute:
    1. Open the Vitamin Personalizer
    2. Proceed with selecting every option in the menu. Ensure that the option displays the screen correctly.
- Post-Execution Actions: The system should correctly display the correct screen to the user based on their input.

2.2.6 Account Login Verification Procedure

- Procedure ID: VP-06 LVP
- Description: The procedure tests the account login verification with the unique ID given from registration. The system should allow the user to login with their ID and deny if either no ID is found or the wrong ID is entered.
- Test Environment: Windows Operating system, a browser that can execute Python program.
- Steps to Execute:
    1. Make sure the user has an account registered in the system with a unique ID.
    2. Log out of the system.
    3. Press the corresponding option to log back into the system.
    4. Enter the name and ID of the account registered.
    5. Press Enter once filled in.
    6. Repeat steps 2-5, except enter an invalid ID and press Enter.
- Post-Execution Actions: The system should allow the user back into the system to input their values. The system should deny the user access to the Vitamin Personalizer with an invalid ID.

2.2.7 PDF Generation Procedure

- Procedure ID: VP-07 PGP
- Description: This test procedure tests the PDF generation for lab results. The user should have the option to print out their results after inputting their values or through the progress check screen.

- Test Environment: Windows Operating system, a browser that can execute Python programs, and a nearby printer.
- Steps to Execute:
  1. Make sure you are either logged into an existing account or have registered one beforehand
  2. The user should have already input their blood work results and be given their results.
  3. Press the number corresponding to "Print results to PDF."
  4. A print screen will pop up with the preview of the printout. Press Print.
  5. Navigate back to the main menu and to Progress check while inputting the corresponding numbers in the prompt.
  6. Repeat steps 3-4 while on the "Progress Check" screen.
- Post-Execution Actions: The system should be able to take the results from the database and print onto a PDF file for the user. The PDF generator should successfully print out 2 physical copies of the user's lab results.

3. Test Execution Report

3.1 Test Results:

3.1.1 Account Registration

- Test Case ID: VP-01 AR
- Test Case Description: Test the account registration feature in the Vitamin Personalizer.
- Test Result: Pass
- Test Log: The Account Registration test passed the functional requirements of the SRS Registration requirements. The account registration worked as intended, allowing the user to enter their information and be given a unique ID.
- Defect ID (if applicable): N/A

3.1.2 Adding New Blood Work Results

- Test Case ID: VP-02 BL
- Test Case Description: Test the lab results creation of the Vitamin Personalizer.
- Test Result: Pass
- Test Log: This test case will pass the functional requirements of the Update Data SRS requirements. Creating lab results worked as intended, allowing the user to enter their blood work values into the database.
- Defect ID (if applicable): N/A

3.1.3 Updating Blood Work Results

- Test Case ID: VP-03 UR
- Test Case Description: Test the update data feature where the user replaces outdated lab results with new ones.
- Test Result: Fail
- Test Log: Although the lab inputs were successfully imputed, the results may differ, as for vitamins that were left blank, it would show the previous results. The database may not work as intended.
- Defect ID (if applicable): DBC_01

3.1.4 High and Low Value Testing

- Test Case ID: VP-04 HL
- Test Case Description: Test the high attention and low attention feature of the Vitamin Personalizer analysis requirement.
- Test Result: Pass
- Test Log: This test will mostly pass the requirements. Although 90% of the vitamins had their values analyzed correctly, some values were either rounded up or down, labeling them incorrectly. Regardless, the user should still seek medical professionals to verify the information given.
- Defect ID (if applicable): N/A

3.1.5 Menu Navigation

- Test Case ID: VP-05 MN
- Test Case Description: Test the menu navigation to check that the menu correctly shows the screens for each option.
- Test Result: Pass
- Test Log:
- Defect ID (if applicable): N/A

3.1.6 Account Login Verification

- Test Case ID: VP-06 LV
- Test Case Description: Test the account login verification by having the user log back into a registered account with a name and ID in the user database.
- Test Result: Pass
- Test Log: The test case was successful in fulfilling the Login SRS requirements, allowing the user back into the system with the correct ID and preventing them from using the system with an invalid ID entered.
- Defect ID (if applicable): N/A

3.1.7 PDF Generation

- Test Case ID: VP-07 PG
- Test Case Description: Test the PDF generator having getting lab results back and through Progress Check.
- Test Result: Fail
- Test Log: Although the user was able to be brought to the print screen from the lab results screen, there seems to be an issue with getting to it through the Progress check screen, where it crashes upon pressing the option to do so.
- Defect ID (if applicable): PDF_02

3.2 Defect Report:

3.2.1 Database Connection through updating values

- Defect ID: DBC_01
- Description: This defect case is about conflicting values between old values and newly input values.
- Severity: Minor
- Steps to Reproduce:

1. Make sure you are either logged into an existing account or have registered one beforehand
2. The user should have already input their blood work results prior.
3. Select the number corresponding to the option “update lab results” and press Enter.
4. For all 50 vitamins, enter your blood work values.
5. For vitamins you want to omit, leave blank.
6. Press Enter once finished.

- Status: Closed
- Fix Version: The fixed version should completely delete the old values by using the user’s ID, and replace them with new values in the new version.

3.2.2 PDF Generation through Progress Check

- Defect ID: PDF_02
- Description: This defect report concerns the user's inability to print their results from the progress check screen.
- Severity: Minor
- Steps to Reproduce:
    1. Make sure you are either logged into an existing account or have registered one beforehand
    2. The user should have already input their blood work results and be given their results.
    3. Navigate through the menu to the Progress Check screen.
    4. Press the number corresponding to “Print results to PDF.”
    5. Check that the print screen displays the current vitamin results and recommendations on the PDF.
- Status: Closed
- Fix Version: The Vitamin Personalizer should be able to print a PDF of the results from the Progress Check screen.

3.3 Traceability Matrix:

ID(s)	**Requirement(s)**	**Test Case(s)**	**Pass/Fail(s)**
VPR-01	The system should allow or deny access depending on the ID and name entered.	**VP-01 AR,** **VP-06 LV**	**Pass**
VPR-02	The system should read the user’s inputs, analyze them, and recommend vitamins based on results.	**VP-02 BL,** **VP-03 UR,** **VP-04 HL**	**Pass**
VPR-03	The system should display the correct screen while navigating through the menus.	**VP-05 MN**	**Pass**
VPR-04	Lab results should be analyzed and correctly displayed.	**VP-02 BL,** **VP-03 UR,** **VP-04 HL**	**Pass**
VPR-05	The system should register the user and give them a unique ID after processing the	**VP-01 AR**	**Pass**

	information.		
VPR-06	The system should label low attention and high attention values based on the vitamin's optimal range and user inputs.	**VP-02 BL,** **VP-03 UR,** **VP-04 HL**	**Pass**
VPR-07	The system should be able to print the user's lab results after receiving the results or through the progress check screen.	**VP-07 PG**	**Fail**
VPR-08	The system should replace old values with new values when updating blood work values.	**VP-03 UR**	**Fail**

4. Test Summary:

4.1 Test Environment:

4.1.1 Hardware

- The hardware will be a Windows operating system or a laptop running Windows. Hardware specifications should be 2GB of storage at most.

4.1.2 Software

- The software is a browser that can run Python and connect to SQL via SQL Connector. It is recommended to run the latest version of the browser, Python, and SQL.

4.2 Overall Test Results:

- **4.1.1 Total Test Cases Executed**: 7
- **4.1.2 Pass/Fail Ratio:** 5 passed, 2 failed
- **4.1.3 Defects Found:** 2 minor defects
- **4.1.4 Overall System Status:** The majority of the system is functional and stable for vitamin recommendations for the general public. There may be slight errors due to rounding of values, but users are advised to consult medical professionals before implementing the recommendations and diet.

4.3 Key Findings:

4.3.1 Strengths

- Demonstrated strong verification functionality with the user registration and login.
- Highlighted effective analysis on vitamin values and recommendations based on the values entered.
- Showcased proficient data storage within separate SQL databases.
- Allows user to update their vitamin values for continued use in the future.

4.3.2 Weaknesses

- The system may accidentally display different recommendations based on rounded values.
- The system showed overlapping old and new data from user inputs when updating their vitamin values.

4.4 Conclusion

- In conclusion, based on the 71% (5/7 test cases) success rate of the test cases, the Vitamin Personalizer still needs more testing and tuning before releasing it for the majority of users. However, because most of the defects were minor, such as fixing access to the PDF Generator and display of old recommendation data, the Vitamin Personalizer's components work and function just as expected and efficiently. It demonstrated strong communication and functionality between the SQL databases and components when it came to registering the user's account information and analyzing the vitamin values for recommendations. Although a medical professional is still highly recommended before pursuing a diet based on the system, the Vitamin Personalizer still provides people with a convenient and accessible way of helping their body's functions and vitamins.