

Software Requirements Specification

Vitamin Personalizer

Nhat-Huy Tran, Jonathan Yunes, Ahmet Ege Aytac
Wellness Wizards
June 22nd, 2025

Table of Contents

- 1. Introduction**
 - a. 1.1 Purpose
 - b. 1.2 Scope
 - c. 1.3 Definitions/Acronyms/Abbreviations
 - d. 1.4 References
 - e. 1.5 Overview
- 2. Overall Description**
 - a. 2.1 Product perspective
 - b. 2.2 Product features
 - c. 2.3 User characteristics
 - d. 2.4 Operating environment
 - e. 2.5 Design and implementation constraints
 - f. 2.6 Assumptions and dependencies
- 3. Specific Requirements**
 - a. Functional Requirements**
 - i. 3.1.1 Login
 - ii. 3.1.2 Register
 - iii. 3.1.3 Menu
 - iv. 3.1.4 Update Data
 - v. 3.1.5 Lab Results
 - vi. 3.1.6 Print
 - vii. 3.1.7 Progress Check
 - viii. 3.1.8 Settings
 - b. Non-functional Requirements**
 - i. 3.2.1 Usability
 - ii. 3.2.2 Performance
 - iii. 3.2.3 Security
- 4. External Interface Requirements**
 - a. 4.1 User Interface
 - b. 4.2 Hardware Interface
 - c. 4.3 Software Interface
 - d. 4.4 Communication Interface
- 5. System Features**
- 6. Other Requirements**

1. Introduction

Purpose

- The purpose of the Vitamin Personalizer Software Requirement Specification (SRS) document is to provide specifications for the developers of the project, which recommends vitamins and diet based on the person's inputted blood test results. This SRS will cover details about the description of the project, functional and non-functional requirements, and interface characteristics for the developers as a reference.

Scope

- The scope of the Vitamin Personalizer software is to give personalized vitamin and diet recommendations based on their blood results, with clear explanations. Users will input their age, height, weight, and blood results, and based on the results and the database of previous results, the software will obtain the vitamins needed for the user. The software will then print out a summary report, with one copy for the person's recommendations and another for database storage for future recommendations. The software will be generalized for the majority of people.

Definitions, Acronyms, and Abbreviations

- **SRS:** The software requirements specifications document, which highlights all of the conditions for a successful system.
- **SQL:** Structured Query Language
- **User:** An individual who interacts with the program to receive diet and vitamin recommendations by entering lab results.
- **API:** Application Program Interface with a 3-layered system where main.py takes user input and shows results, the Lab Analyzer checks values and gives recommendations, and SQLite handles saving and loading the data.
- **Vitamins:** Substances necessary for growth, development, and nutrition. Based on the blood work, the software recommends the person the necessary vitamins needed for growth.

References

- **IEEE.** (1998). IEEE Std 830-1998, IEEE recommended practice for software requirements specifications. IEEE.
- **U.S. Department of Health and Human Services.** (n.d.). *Vitamin and mineral supplement fact sheets*. NIH Office of Dietary Supplements.
<https://ods.od.nih.gov/factsheets/list-VitaminsMinerals/>
- **U.S. Department of Health and Human Services.** (n.d.-a). *Office of dietary supplements - nutrient recommendations and databases*. NIH Office of Dietary Supplements.
<https://ods.od.nih.gov/HealthInformation/nutrientrecommendations.aspx>

Overview

- The rest of the SRS document will cover the specifications for the Vitamin Personalizer software, including functional and non-functional requirements, interface characteristics, and other additional material within the project.

2. Overall Description

Product Perspective

- The vitamin personalizer is generalized for many types of people which allow more insight such as different diets and vitamin usages from them to accurately recommend them to the user. The product uses previous records to help with predictability and continues to improve thanks to the registered users and allowing them to update their recommendations.

Product Features

- Users can register their name, age, sex, height, and weight for a Unique ID. It is then stored within a database for future blood work.
- Users already registered can continue updating their vitamin and diet recommendations the next time they update their blood work results input.
- The product takes the user's blood work input and previous blood work input and compares each value to see whether they're normal, high, or low for vitamin predictions.
- The product uses previous records from multiple databases to accurately predict the recommended vitamins based on past reports.
- The product will display the key values within the person's blood, such as minerals and vitamins within the bloodstream, alongside symptoms the person may be experiencing, and explanations either by the product through input.
- The product stores the user's blood work data throughout so that they can use it for progress checks and graph creations to view their results over time.
- A document containing the user's registered information, date, recommended vitamins, and short description is then printed and handed to the user either physically or electronically. A 6-month future date for the user to update their vitamin recommendation will be printed on the document.

User Characteristics

- The targeted users will mostly be individuals with dietary restrictions, with secondary users being older people and people with fitness like athletes. Those with dietary restrictions may not be able to take in some vitamins due to their diet, and so a doctor can take their blood work and recommend them based on their registered information and dietary restrictions.

Operating Environment

- Operating system: Windows 10, using a Python program with matplotlib and SQL database connector.

Design and Implementation Constraints

- Databases full of previous reports may contain private or unauthorized information
- Databases may contain either inconsistent data or missing data, which could potentially lead to incorrect vitamin recommendations
- Some vitamins are more common than others, meaning that there may not be any record of the rare vitamins needed, resulting in loose, incorrect, or missing recommendations.
- A password system is implemented to prevent a security risk of brute force of IDs

Assumptions and Dependencies

- The product depends on user information such as age, gender, weight, height, lab results, and blood test results to function.
- The product also depends on previous reports from multiple databases to make correct predictions.

3. Specific Requirements

Functional requirements:

- **Login**
 - Upon starting up, the user should have the option to log in with their unique ID.
 - If the user does not remember their unique ID, they can press the "forgot ID" button and enter their registered email for the software to send them their ID.

- If the login was correct, it should direct the user to the menu where they are given the option to input their results or check their progress.
 - If the login was incorrect, the user is given a reminder to try again. Upon 3 unsuccessful attempts, the user cannot attempt again until an hour has passed.
 - There should be an option for the user to register underneath the login if they do not have an account for the personalizer.
- **Register**
 - If the user is new, they can press the register button, where they can enter their username, email, full name, age, sex, height, and weight.
 - There should be an age limit to where the user can only input a proper age within a certain range. If the user inputs something like 121 for age, for example, it will give the user an error and instruct them to try inputting a proper age. This is the same for height and weight, with different values.
 - If the username, email, or full name matches with an already registered user on the app, there should be an error in which any of the 3 is already taken and that the user should use a different input.
 - Once filled out, the software should give the newly registered user a unique ID that they can use to log back in if needed.
- **Menu**
 - After a successful login attempt, the software should direct the user to the menu where they have the option of either pressing “Update Data,” “Progress Check,” ”View Lab Results,” and “Settings.”
 - The software should direct the user to their stated page or menu when pressed.
 - The software should include a log-out button for the user once they are finished with their activity.
- **Update Data**
 - The software should tell the user to input their blood work data, such as white blood cells, red blood cells, glucose, and more.
 - The software should allow the user to re-enter their blood work results if updated.
 - The software should note the user of the time and date that the blood work result was inputted every time a user has successfully entered their data.
 - Each component of the blood work data should have an optimal range where it is considered normal. Any number outside of that range should be considered requiring high attention or low attention.
 - Once the data has been inputted, the software should save the input and store it within a database. The user should receive a “lab test results have been successfully inputted!” message and be directed to the lab results after inputting their data.
- **Lab Results**
 - The software should output the results for the user once they have input their blood work results.
 - The software should compare each input for each blood work component and compare them along with previous records.
 - If a component’s input is within the optimal range, it will be labeled as normal.
 - If a component’s input is outside of the range, either too high or too low, it will be labeled as abnormal.
 - For abnormal components, the software should provide the status and the recommended diet and supplement.
 - For normal components, the software should print out the status.
 - The lab test results should be stored within the personalizer’s database for the user’s progress check and component comparisons.

- After the results have been shown, the user has the option to either print out the results or enter another blood work data input. Otherwise, they should be directed back to the main menu.
- **Print**
 - If the user clicks on the print button at the bottom left of the lab results screen, they will be directed to the print screen, where they can print the lab results in a PDF format or on physical paper.
 - The software should have another print button in the progress check screen, where the user can view their previous results from the lab test.
 - The software should give the user the option to print multiple copies or multiple test results.
- **Progress Check**
 - If the user presses the “Progress Check” button in the main menu, they will be directed to the progress check screen, where they can create, view graphs, or view previous results.
 - The software should allow the user to create a line or bar graph and track their progress overtime by assigning Time as the x value and any blood work component as the y value.
 - The software should display the optimal range of each blood work component on the graph to help the user keep track.
 - The software should automatically create the graph using the lab test data inputted from the user over time. The graph should show either an increase or decrease in the blood work component over time in a line or bar graph.
 - The software should allow users the view the graphs they’ve created and the previous lab test results.
 - The software should allow the user to print out either their previous lab test results or graphs they’ve created in the progress check screen.

Non-functional:

- **Usability**
 - The system should be easy to use, read, and easy to understand, with intuitive navigation and clear prompts.
- **Performance**
 - Performance requirements include fast response times under 2 seconds for data entry and 5 seconds for analysis completion.
- **Security**
 - Security requirements include local data storage only, input validation for all lab values, and user privacy protection through unique identification systems.
 - The system must be maintainable with a modular code architecture and an extensible design for adding new tests.

4. External Interface Requirements

User interfaces

- **Login/Update/Register**
 - First-time users can register their username, age, and email, and will be given a unique ID for the program. Continuous users can login with their username/email and password and can update their account information if necessary.
- **Password**
 - Once a user is registered in a database, they can create and enter a unique password for their records to prevent brute-force attacks on IDs.

- **Blood work/Lab Input**
 - After acquiring results from blood work, users can input the information from the blood work, such as a person's blood sugar level, iron levels, and more.
- **Vitamin Personalizer**
 - Users have the option to either click on Vitamin Personalizer or a Diet Personalizer after inputting their lab values and considering previous lab records. Vitamin Personalizer gives the user their recommended vitamins, along with supplements and dosages for deficiencies.
- **Diet Personalizer**
 - Just like Vitamin Personalizer, Diet Personalizer gives users a recommended/personalized diet based on lab values alongside a recommended diet to exercise it.
- **Dictionary**
 - In the menu, users can view the vitamin dictionary, which is a list that explains each vitamin, its optimal level, and its health benefits. Explanation also includes the effects of deficiency and overconsumption.
- **Progress Tracker**
 - All registered users have access to a personal progress tracker that records the user's inputs over time. The progress tracker can be customized to record the progress of any vitamin or mineral, such as the consumption of vitamin D(Y) over time between each lab input date (X). The progress tracker also allows the user to save multiple graphs to help the user keep track of everything.
- **Printer option**
 - The doctors can either send an email to the user or print out a document containing their information and results to the patient. Users also have the option to print out graphs saved from the progress tracker.

Hardware interfaces

- The software should work on a computer, laptop, or other devices that can run SQL and Python.

Software interfaces

- The software should run on the latest version of Windows 10.
- The software should connect to an SQL database of previous records to increase the accuracy of the vitamin recommendations.
- The software should generate reports and graphs in the progress tracker interface of the software, using lab inputs as data, time/date as X values, and vitamins/minerals as Y values.
- The software can print the patient information and results on a PDF to be printed for the patient and stored in the database for later use.

Communication interfaces

- The imputed information from the patient, such as name and age, will be transferred to a database for registered users for continuous blood work.

5. System Features

- The system features include a user registration system for creating accounts with personal information. Secure ID authentication allows users to login and access their data. Lab data input supports entry of medical test values across multiple categories. Value range analysis compares entered values against medical reference ranges to classify results. Diet recommendation generation provides personalized dietary advice based on lab values. Vitamin recommendation generation suggests specific supplements with dosages for deficiencies. The vitamin dictionary

explains what each vitamin is for and its health benefits. The results display interface presents organized findings with clear visual indicators. Historical data storage maintains previous test results with timestamps. Progress tracking capability allows users to compare results over periods. Report generation functionality enables downloading analysis results as PDF or Word documents. Data update management allows users to modify lab values while preserving records.

6. Other Requirements

- Any additional considerations, such as legal, regulatory, or compliance needs.
- The system needs to show a clear warning that everything is just for information, not medical advice. Users should know they need to talk to a real doctor before making any health decisions. To make sure everything is safe and accurate, a health expert should review the data and recommendations.
- There should also be a user agreement that people have to accept before using the system. It should explain that we're not responsible for medical outcomes, and the system doesn't give prescriptions. All supplement suggestions are for things that can be bought over the counter, so no need for a prescription to use the app easily.