“SuppleMate” App
Software Quality Assurance Plan
(SQAP)

Team:

- Gregy Thomas Kokkaparampil – Project Manager
- Shuo Li– Assistant Project Manager/ Software Engineer
- Siddharth Santhanakrishnan – Software Engineer
- Wone Eui (Ella) Jung – Software Engineer

Table of Contents

1. Purpose

The Software Quality Assurance Plan (SQAP) is the document to control the integrity of the code being written by the Flowlab team for SuppleMate App. It is based on the Software Requirement Specification (SRS) and Software Project Management Plan (SPMP), to track processes and ensure the applications are meeting client demands.

The SuppleMate is designed to help users manage their dietary supplements and vitamins effectively. It allows users to input and track their supplement intake, compute the exact dosage required each time, set dosage reminders, and receive personalized nutritional/ supplement recommendations based on their health goals and nutritional needs. The app provides insights into the benefits of each supplement, tracks expiration dates, and offers a comprehensive overview of the user's overall supplement regimen.

2. Reference Documents

[1] IEEE Standard Software Quality Assurance Plan (Std 7201998)

[2] SuppleMate App Software Requirements Specification (SRS)

[3] SuppleMate App Software Project Management Plan (SPMP)

3. Management

The team consists of four members. The project manager and assistant project manager will represent QA for discussions and relay information to Dr. Concepcion when needed, as well as manage and schedule meetings with team members. Each software engineer is assigned to different functions, and maintains communication between project and project manager. The whole team will gather all the information of the project together to make sure to test it properly.

4. Documentation

Software Requirements Specifications (SRS)

Software Project Management Plan (SPMP)

Software Quality Assurance Plan (SQAP)

Each team member is responsible for monitoring on a weekly basis to track progress and perform incremental testing of various functions (if possible). When the prototype is available, all team members will execute strategic testing of the prototype.

Testing will be designed based on SRS. There will be specific functionality requirements which will direct test cases and ensure the application is performing as specified.

Each function of the application will be thoroughly tested with intent to make the application malfunction. All tests will be recorded by each team member for repeatability purposes and to maintain records.

5. Standards, Practices, Conventions, and Metrics

5.1 Coding Standards

The following links contain coding standards for the languages used to develop the web application:

Java Google Style Guide:
https://google.github.io/styleguide/javaguide.html

XML Google XML Document Format Style:
https://google.github.io/styleguide/xmlstyle.html

Providing and enforcing coding style guides will ensure that the project maintains readable code. Code that can be easily read makes quality assurance easier for all team members and also keeps the development team organized, which promotes efficiency and productivity.

5.2 Development Practices

Since the team will be checking their respective repositories on a weekly basis, it's important that all team members are consistently committing code on Github. They should follow these general practices:

- Be knowledgeable about Github and how to add, commit and push code.

- Commit code often, but only when the code is functional.

- Add useful and detailed messages to each commit.

- Follow coding standards provided.

5.3 Software Metrics

5.3.1 Define "Fault" or "Defect": We must clearly define what the team considers a fault or defect. We need to check if there is any deviation from the expected behavior, or if there are specific criteria for categorizing issues

5.3.2 Recording and Logging: Specify how the team will record and log defects. Ensure that there is a standardized and consistent process for capturing this information.

5.3.3 Default Recording Log: The statement mentions collecting data from the "default recording log." It's essential to define what this default log is and ensure that it captures all relevant information about defects.

5.3.4 Regular Review and Analysis: Establish a process for regularly reviewing and analyzing the collected data. This could involve periodic inspections of the defect logs to identify patterns, trends, and areas for improvement.

5.3.5 Continuous Improvement: Use the insights gained from the metric to drive continuous improvement efforts. If the defect density is high, it may indicate areas of the codebase that need attention or improvements in the development process.

6.Reviews and Audits

A review will be conducted together by all team members for the web application. With the SRS in hand and the completed prototype, the application will be compiled and run. It will pass the audit for its prototype version if its functionality matches the SRS and the program runs smoothly without too many issues.

7. Test

In this section are specific requirements that must be satisfied in order to pass testing for each phase of development. These requirements are supplied by the SRS.

7.1 User Profile creation

Prototype 1:

Check Register and Login Functionality:

- User Registration
  - Register user information
  - Update personal health information and supplement information
  - Upload all personal information in Database
- Login
  - User can log in and safely log out
- Update all user profile information
  - Reflect updated information in Database

7.2 Supplement reminder

Prototype 1:

Check Reminder accuracy

- Reminder
  - Send reminder email to user to notify schedule and intake amount

        - Shared schedule on user’s outlook or google calendar

7.3 Supplement Intake Calculation and Scheduling

Prototype 2:

Check Supplement amount calculation and Scheduling

- Check accuracy of recommended amount
    - Calculate intake amount
    - Check Max intake by age and gender
- View supplement Dashboard
    - Show currently taking supplement
    - Show recommended amount
- Scheduling
    - Check schedules for supplement intake based on recommended amount
    - Work update for schedule when changes of product or stop taking supplement

8. Problem Reporting and Corrective Action

All problems dealing with a fault of the applications will be recorded and reported to Project Manager. A testing report will be created clearly indicating whether they have passed or failed. This report will contain the test cases and descriptions of the bugs found for each case. Project Manager will communicate the results of the tests to all team members. Software engineers will correct the problems and resubmit a new updated version of the product for further testing. The testing report will look something like this:

SuppleMate Prototype1	
Test Case Description	**General Information**
	Name: name of member doing TC
	Date: Date of doing TC
	Pass/ Fail: Pass / Fail check
	Resolved?: Overall check
	Comments: How it failed
Standard Code Compliance	**Compliant/ Not Compliant**

9. Tools, Techniques, and Methodologies

Team members will be using Google Drive to help manage the software process. Google Drive will allow us to assign the issues to the team member in charge of each function. Also, the Project manager will compare the results to the test cases specified in the SRS.

10. Code Control

All code written and updated by respective software engineers will be available in their Github repositories. Each team member has read and write access to all code and documents corresponding to the development process of their functions. All software engineers will also look into the source code of each function to see if the software engineers who wrote the code are following the coding standards.

11. Media Control

All source code and executable files will be backed up to prevent any accidental or deliberate loss of data. There will be various methods used for backing up the data such as a USB flash drive, external hard drive, personal computers, Google drive and any other form of backup available. Backups should be done regularly by software engineers to avoid any loss of data.

12. Records Collection, Maintenance, and Retention

All test cases of the project will be documented. This will ensure that all, if any, errors that are recorded during testing of a project are to be fixed by the next stage of development. The first collection of test cases will be kept as a reference for testing in the future. And the collection of test cases will be kept to maintain high quality assurance.

13. Training

All team members are required to have some familiarity with web design. Any additional training will be established by the project manager, listed within their respective SPMPs and Github.

Using Github:
https://docs.github.com/en/get-started/quickstart/hello-world

Using IntelliJ Idea Ultimate 2023:

https://www.jetbrains.com/help/idea/feature-trainer.html

Using Postman:

https://learning.postman.com/docs/introduction/overview/

Using Figma:

https://www.figma.com/resource-library/prototyping/

14. Risk Management

In the event of any unforeseen circumstances, it is required that all files will be reviewed and backed up in Media Control to solve any problems that may have occurred. In the event of a team member's absence due to any reason, the other members will take over the test cases assigned to the absent member and test the application.