Software Requirements Specification

for

Eat what? Application

Version 1.0 approved

Prepared by Jacky Xu, Allen Liu, Chris Gai

XXX Company

February 27, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Whole Group	27/2/2022	Initially established	V1.0

1. Introduction

1.1 Purpose

In response to the need for diet control for fat loss or fitness people, we have developed an Android application to provide them with fitness guidance by recording and predicting the physical changes brought about by diet.

The application is currently in its infancy, with the version number V1.0, and the software requirement specification analysis version number V1.0.

1.2 Document Conventions

The Android app is a separate installation package, which requires the use of mobile phone hardware and network connection to achieve all functions.

All functions appearing in the following text are based on a unified software specification, safety specification and communication specification.

1.3 Intended Audience and Reading Suggestions

Users can choose to read some of the content that they have questions or are interested in.
The tester should read from the beginning to provide a full experience report.
Developers and related project personnel should pay attention to the actual functions and non-functional parts of the software, and then read the parts related to the development work and process.

1.4 Product Scope

Eat what? is a public welfare application designed to meet the needs of fitness people for diet control, and to provide help for these people who are controlling their diet.

1.5 Reference

Krebs, P., & Duncan, D. T. (2015). Health app use among US mobile phone owners: a national survey. JMIR mHealth and uHealth, 3(4), e4924.

Tang, J., Abraham, C., Stamp, E., & Greaves, C. (2015). How can weight‐ loss app designers' best engage and support users? A qualitative investigation. British journal of health psychology, 20(1), 151-171.

2. Overall Description

2.1 Product Perspective

This software is an enhancement version of current diet application in mobile. Although some current diet applications have matured development in their functions. However, as a diet application facing amateurs it leaks basic features to attract user and encourage user to keep using it. So, comparing with traditional diet application, the application proposed in this document would be more user-friendly, entertain and appealing.

2.2 Product Functions

The software contains five basic functions. User data record system: User will enter their basic information and target weight. User also allow to change their information and target weight. These data will be applied to calculate user' s daily consuming and diet strategy. Calories record system: User could record their daily calories embedded time record component is recommended. Food calories menu: A menu user could search their consumed food' s calory a convenient search method and a connection to Calories record system is recommended. Calories consuming reflect system: This is a core system of this application. This system aims at encouraging user to record their calories frequently. The calories user has recorded would be reflect into an entertainment approach which would be discussed later. Diet strategy system: Application will display recommend strategy for user to diet. The system overview figure is displayed below.

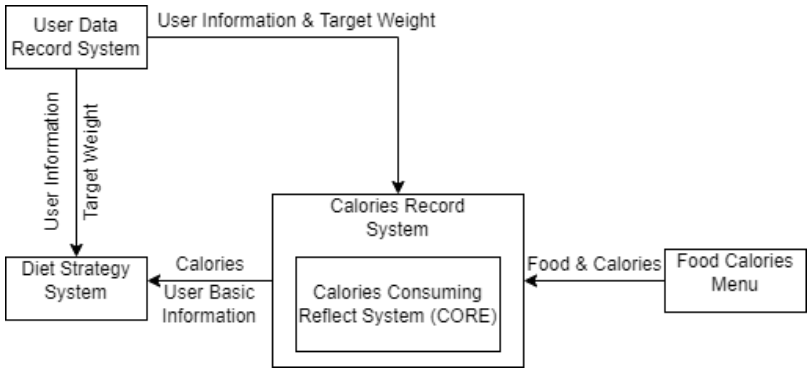

Fig 1: System Overview

2.3 User Classes and Characteristics

This application is developed for 20 to 50 years old people who want to diet. Most of the user would be amateur which means that they may have low tolerance and patient in using the system. Thus, simplifying the procedure of recording calories and enhancing the entertainment would be the core issues.

2.4 Operating Environment

2.4.1 Software environment: Android 4.0 and above.

2.4.2 Hardware environment: dual-core 2GHz (and above) + 4G memory (and above) + 32G hard disk capacity (and above)

2.5 Design and Implementation Constraints

2.5.1 Simplifying operation

Except searching food calories, rest of the operation to access each function should be no more than four steps. Buttons and information arrangement should be clear and well arranged. Also, the time latency should be no more than 1 seconds. This means high usability.

2.5.2 Color

The use of light color and the number of colors should not be too much. Also, for the security of user, avoid flicker, strong irritating color, and any possibility to trigger photosensitive epilepsy.

2.5.3 Language Requirements

The application supports Chinese and English version, and allows user to change the languages.

2.5.4 Security consideration

No user information would be disclosure.

2.6 User Documentation

2.6.1

The application should provide a user manual to guide user how to enter their information, record their calories, using entertainment system, and access to each system.

2.6.2

An bug report and help system should be provided.

3. External Interface Requirements

3.1 User Interfaces

The software's user interface is designed by Adobe Illustrator and produced by Kivy. Environment requirements for Adobe Illustrator and kivy: MacOS 11, multi-core Intel processor, x86-64 architecture, 8 GB RAM, 512 GB SSD, python3.8.10. It requires an internet connection. An Adobe ID and license agreement are required to enable and use Adobe Illustrator. This product integrates with or allows access to certain Adobe or third-party hosted online services. Browsers should use HTTP when accessing the protocol, use port number 80, and require an IPv4 logical system address.

3.2 Hardware Interfaces

Since applications must run on the Internet, all hardware that needs to connect to the Internet becomes the hardware interface to the system, such as modems, WANs, and Ethernet crossover cables. All client-side components must execute on phones with the Android 4.0 system and above.

3.3 Software Interfaces

The software interface should follow the SEO model for rendering data objects. The interface must be able to crawl the internet to get the data of the target website or its metadata in HTML meta tags. Then the interface must be able to connect to the local database to store web data for future processing. In the end, it must be able to provide the user with relevant searching results. Source and destination formats for data must include XML and may also include: Extensible Stylesheet Language Transformation (XSLT), JavaScript Object Notation (JSON), Comma Separated Value (CSV), and American Standard Code for Information Interchange (ASCII).

1. The software interacts with the user interface, allowing the user to drag or select one or more images.
2. The software provides a graphical conversion interface that allows users to translate the picture definition and the picture itself.
3. The software provides a database interface driven by the SQLite3 module, allowing developers to write user behaviors into the database.
4. The software and hardware interface interaction allows the system to collect, analyze and feedback user data.
5. The software provides a recommendation interface based on a collaborative recommendation model and machine learning, and makes personalized phrase recommendations for users to determine the priority of subsequent associations.
6. The software provides a monitoring interface, allowing the superior to monitor the subordinate user interface, and the superior has full control authority.
7. At the same time, based on the Android11 operating system, the software will call the application layer interface for partition storage, save the proprietary application files in the exclusive location, and obtain the reproduction permission and background access permission, and automatically locate the information.

3.4 Communications Interfaces

The communication architecture must follow the client-server model. Communication between the client and server should utilize a REST-compliant web service and must be served over HTTP

Secure (HTTPS). The client-server communication must be stateless. A uniform interface must separate the client roles from the server roles.

4. System Features

4.1 System Feature 1

4.1.1 Description and Priority

Users can simulate their own eating process by "drag and drop" the cartoon food pictures below. The app records the user's caloric intake and shows the user how reasonable the diet is through the physical changes of the cartoon characters (fat or thin).

4.1.2 Functional Requirements

Most diet recording software calculates calories in text form. But the plain text interface will reduce the user's enjoyment of using the software. After losing the fun of using the software, users may open the software less frequently, and may even forget to record and plan their daily diet. This is what we don't want to see. Using a dynamic figure can let users have interactive fun when using the software, so that the use process is not boring. Let users use "game"-style software rather than "task"-style software.

4.2 System Feature 2 (and so on)

4.2.1 Description and Priority

Whether it is the simulation process of eating or exercising, we will provide users with certain suggestions as a reference to help users develop healthier living habits.

4.2.2 Functional Requirements

The software will analyze the needs of users in a targeted manner. For example, users who are overweight or thin will get completely different exercise recommendations. Controlling your diet is not the only factor in getting a good body, exercising is the best way.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The basic function of the software is to record the diet and give relevant suggestions to the user after getting the results. Since the main user group is the diet-controlled exercise crowd, all the interfaces are in light green. Light green can soothe the mood, and its light tone can also highlight

the dynamic figure to enhance the user's visual experience. The text part also uses light brown with a softer color temperature, which does not give users an overly stimulating visual impact.

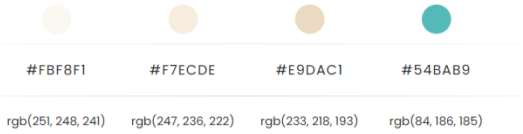

Fig 2: Color

5.2 Safety Requirements

This app needs to be logged in to use it. Once a user is logged in to their account, they can stay online all the time. Users can add their own body data and choose whether to upload it to the cloud. If changes are required, you will need to log in to your account again and verify your identity.

5.3 Security Requirements

We strictly protect and store user data. We will not disclose any user's private information to the outside world without reason. User data is uniformly stored on our local servers instead of cloud servers, which are more easily leaked.

6. Appendix

6.1 Login scene

6.2 Record scene

6.3 Main scene