

Fred++

Deliverable #1: SRS

SE 3A04: Software Design II – Large System Design

Group 9:

Kunal Shah

Gabriel Castagner

Victor Velechovsky

Josh Mitchell

1 Introduction

This section of the SRS should provide an overview of the entire SRS.

1.1 Purpose

- a) Delineate the purpose of the SRS
- b) Specify the intended audience for the SRS

1.2 Scope

- a) Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.)
- b) Explain what the software product(s) will, and, if necessary, will not do
- c) Describe the application of the software being specified, including relevant benefits, objectives, and goals
- d) Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist

1.3 Definitions, Acronyms, and Abbreviations

- a) Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS

1.4 References

- a) Provide a complete list of all documents referenced elsewhere in the SRS
- b) Identify each document by title, report number (if applicable), date, and publishing organization
- c) Specify the sources from which the references can be obtained

1.5 Overview

- a) Describe what the rest of the SRS contains
- b) Explain how the SRS is organized

2 Overall Description

This section describes the general factors that affect the application and its requirements.

2.1 Product Perspective

Fred++ is an Android application and as such its functionality relies on the Android operating system as well as the touch screen input and display/audio output of the hardware (phone, tablet, etc) on which it is running.

This application will be responsible for receiving input from the hardware's touch screen, processing that input as it relates to the current state of the system, and outputting any necessary visual and auditory data to the hardware's display and speakers.

2.2 Product Functions

Fred++ will allow users to interact with a digital character named Fred and influence his health decisions. They will be able to view Fred's various anatomical systems to see how exactly each decision they make is affecting him. In addition, various health metrics will be available to quantitatively measure Fred's well-being. If the user repeatedly chooses poor health choices for Fred, he may suffer adverse effects which grow in severity. These effects may be reversible, and it will be the user's responsibility to make Fred well again.

2.3 User Characteristics

Fred++ is intended to be used by those with a working knowledge of Android device operation, as well as a basic knowledge of healthy vs unhealthy lifestyle decisions (eg. diet & exercise).

Users need not have a formal education in computer science nor a medical degree to operate this program.

2.4 Constraints

Factors that constrain the development of Fred++ fall into 2 categories: technology and time.

Technological constraints consist of aspects like hardware processing power, maximum screen resolution and battery life. These obstacles must be taken into account, so as to not overextend the practical limitations of the deployment environment.

Time constraints also play a role, as this application must be finished before a pre-determined deadline. As such, some more in-depth features may not be possible to implement in time.

2.5 Assumptions and Dependencies

As Fred++ is an Android application, a reasonable assumption to make is that a sufficiently up-to-date version of the Android operating system will be available on the designated hardware (phone, tablet, etc) on which the application will run.

2.6 Apportioning of Requirements

Due to aforementioned time constraints, some subtleties within the various ways the user can interact with Fred may need to be delayed until future versions. For example, limiting the number of different foods that are available to feed Fred.

3 Functional Requirements

This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every

input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output.

You normally have two options for organizing your functional requirements:

1. Organize first by *business events*, then by *viewpoints*
2. Organize first by *viewpoints*, then by *business events*

Choose the one which makes the most sense.

For example, if you wish to organization by business events:

BE1. Business Event

VP1.1 Viewpoint

- i. Requirement
- ii. Requirement
- iii. ...

VP1.2 Viewpoint

- i. Requirement
- ii. Requirement
- iii. ...

VP1.3 ...

BE2. Business Event

VP2.1 Viewpoint

- i. Requirement
- ii. Requirement
- iii. ...

VP2.2 Viewpoint

- i. Requirement
- ii. Requirement
- iii. ...

VP2.3 ...

OR, if you wish to organization by viewpoints:

VP1. Viewpoint

BE1.1 Business Event

- i. Requirement
- ii. Requirement
- iii. ...

BE1.2 Business Event

- i. Requirement
- ii. Requirement
- iii. ...

BE1.3 ...

VP2. Viewpoint

BE2.1 Business Event

- i. Requirement

- ii. Requirement
 - iii. ...
- BE2.2 Business Event
 - i. Requirement
 - ii. Requirement
 - iii. ...
- BE2.3 ...

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

- LF1. The graphical user interface shall be visually appealing
- LF2. The graphical user interface shall be colorful and

4.1.2 Style Requirements

- LF1.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

- UH1.

4.2.2 Personalization and Internationalization Requirements

- UH1.

4.2.3 Learning Requirements

- UH1. The application shall be easy to use by a so-called "typical" user.

4.2.4 Understandability and Politeness Requirements

- UH1. The application should be language ambiguous, limiting written language in favor of graphical icons, in order to be usable globally, regardless of language.

4.2.5 Accessibility Requirements

- UH1.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

- PR1. The application will load on the majority of devices within 5 seconds of the user clicking the application icon.
- PR2. The GUI thread shall not be blocked/frozen at any time for more than 3 seconds.

4.3.2 Safety-Critical Requirements

- PR1. The application should not cause the phone CPU to overheat.
- PR2. The application should not "drain" a phone's battery more than 15% faster than standard usage on that device.
- PR3. The graphics of the application should not be vomit inducing or seizure inducing, or otherwise cause any physical discomfort in any of its users.
- PR4. System rollbacks, updated at least weekly, should be available and deployable within 24 hours of a fatal disaster in the system.

4.3.3 Precision or Accuracy Requirements

- PR1.

4.3.4 Reliability and Availability Requirements

- PR1. The mean time between failures should be less than 5000 use hours

4.3.5 Robustness or Fault-Tolerance Requirements

- PR1.

4.3.6 Capacity Requirements

- PR1.

4.3.7 Scalability or Extensibility Requirements

- PR1. The application should be client-side based, only relying on a server to download updates.
- PR2. The application should be highly scalable with an increasing user base. The scaling complexity should be, at worst, linear with respect to the size of the user base

4.3.8 Longevity Requirements

- PR1.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

- OE1. The system will run on Android devices with a minimum OS version of Android Lollipop.

4.4.2 Requirements for Interfacing with Adjacent Systems

- OE1. Time will only "pass" in the application when it is running.

4.4.3 Productization Requirements

- OE1. The system should function without an Internet connection.
- OE2. The application should be sold on the Google Play Store for \$3 USD

4.4.4 Release Requirements

- OE1. The application should be in a functional public-release state by April 6, 2017.
- OE2. The application should be publiclyavailable in the Google Play Store.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

MS1. The documentation of the system should remain relevant and up to date, being updated within 5 days of any major changes made to the system.

4.5.2 Supportability Requirements

MS1.

4.5.3 Adaptability Requirements

MS1.

4.6 Security Requirements

4.6.1 Access Requirements

SR1.

4.6.2 Integrity Requirements

SR1.

4.6.3 Privacy Requirements

SR1.

4.6.4 Audit Requirements

SR1.

4.6.5 Immunity Requirements

SR1.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

CP1.

4.7.2 Political Requirements

CP1.

4.8 Legal Requirements

4.8.1 Compliance Requirements

LR1.

4.8.2 Standards Requirements

LR1.

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
 - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.