HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

Version 1.0

<Project Name>

Subject: <Name of subject>

<Group Number>

<List of participants>

Hanoi, <month, year>

<All notations inside the angle bracket are not part of this document, for its purpose is for extra instruction. When using this document, please erase all these notations and/or replace them with corresponding content as instructed>

<This document, written by Asst. Prof. NGUYEN Thi Thu Trang, is used as a case study for student with related courses. Any modifications and/or utilization without the consent of the author is strictly forbidden>

Table of contents

1 Introduction

<The following subsections of the Software Requirements Specifications (SRS) document should provide an overview of the entire SRS. The thing to keep in mind as you write this document is that you are telling what the system must do – so that designers can ultimately build it. Do not use this document for design!!!>

1.1 Objective

<Identify the purpose of this SRS and its intended audience. In this subsection, describe the purpose of the particular SRS and specify the intended audience for the SRS>

1.2 Scope

<In this subsection:

(1) Identify the software product(s) to be produced by name
(2) Explain what the software product(s) will, and, if necessary, will not do
(3) Describe the application of the software being specified, including relevant benefits, objectives, and goals
(4) Be consistent with similar statements in higher-level specifications if they exist

This should be an executive-level summary. Do not enumerate the whole requirements list here>

1.3 Glossary

<Listing and explaining the terms appearing in the software's profession and this documents. Any assumption of the reader's prior knowledge or experience on the subject is ill advised>

1.4 References

<Listing the referenced material used in this documents, including the one related to the project>

2 Overall Description

< Describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in section 3, and makes them easier to understand. In a sense, this section tells the requirements in plain English for the consumption of the customer. Section 3 will contain a specification written for the developers>

3 Specific Requirements

<This section contains all the software requirements at 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. The following principles apply:

(1) Specific requirements should be stated with all the characteristics of a good SRS
- *correct*
- *unambiguous*
- *complete*
- *consistent*
- *ranked for importance and/or stability*
- *verifiable*
- *modifiable*
- *traceable*

(2) Specific requirements should be cross-referenced to earlier documents that relate

(3) All requirements should be uniquely identifiable (usually via numbering like 3.1.2.3)

(4) Careful attention should be given to organizing the requirements to maximize readability (Several alternative organizations are given at end of document)

Before examining specific ways of organizing the requirements it is helpful to understand the various items that comprise requirements as described in the following subclasses. This section reiterates section 2, but is for developers not the customer. The customer buys in with section 2, the designers use section 3 to design and build the actual application.

Remember this is not design. Do not require specific software packages, etc unless the customer specifically requires them. Avoid over-constraining your design. Use proper terminology:

The system shall… A required, must have feature

The system should… A desired feature, but may be deferred til later

The system may… An optional, nice-to-have feature that may never make it to implementation.

Each requirement should be uniquely identified for traceability. Usually, they are numbered 3.1, 3.1.1, 3.1.2.1 etc. Each requirement should also be testable. Avoid imprecise statements like, "The system shall be easy to use" Well no kidding, what does that mean? Avoid "motherhood and apple pie" type statements, "The system shall be developed using good software engineering practice"

Avoid examples, this is a specification, a designer should be able to read this spec and build the system without bothering the customer again. Don't say things like, "The system shall accept configuration information such as name and address". The designer doesn't know if that is the only two data elements or if there are 200. List every piece of information that is required so the designers can build the right UI and data tables>

4 Supplementary specification

<Presenting other requirements if necessary, including non-functional requirements such as performance, reliability, usability, and supportability; or other technical requirements such as database system, used technology…>

4.1 Functionality

<Functional requirements that are general to many use cases>

4.2 Usability

<Requirements that relate to, or affect, the usability of the system. Examples include ease-of-use requirements or training requirements that specify how readily the system can be used by its actors>

4.3 Reliability

<Any requirements concerning the reliability of the system. Quantitative measures such as mean time between failure or defects per thousand lines of code should be stated>

4.4 Performance

<The performance characteristics of the system. Include specific response times. Reference related use cases by name>

4.5 Supportability

<Any requirements that will enhance the supportability or maintainability of the system being built>

4.6 Other requirements

<Descriptions of other requirements are located here>