

Software Requirements Specification (SRS) and Design Document Template

1. Introduction

The introduction of the SRS should provide an overview of the entire SRS.

It should contain the following subsections:

- a) Purpose
- b) Scope

1.1 Problem Description

This subsection should

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

1.2 Scope

This subsection should

- 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;

2. Overall description

This section of the SRS should describe the general factors that affect the product and its requirements.

2.1 Product perspective

2.1.1 A block diagram showing the major components of the larger system, interconnections, and external interfaces

2.1.2. Hardware Requirements

2.1.3. Software Requirements

2.2 Product functions

This subsection of the SRS should provide a summary of the major functions that the software will perform.

3. Specific 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.

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.

3.1 External interfaces

This should be a detailed description of all inputs into and outputs from the software system.

3.2 Functional Requirements

Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as “shall” statements starting with “The system shall...”

3.3 Performance requirements

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

- a) The number of terminals to be supported;
- b) The number of simultaneous users to be supported;
- c) Amount and type of information to be handled.

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

3.4 Design constraints

Specify design constraints imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

3.5 Logical database requirements

This should specify the logical requirements for any information that is to be placed into a database. This may include the following:

- a) Types of information used by various functions;
- b) Frequency of use;
- c) Data entities and their relationships;
- e) Integrity constraints;
- f) Data retention requirements.

3.6 Software system attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified.

3.6.1 Reliability

This should specify the factors required to establish the required reliability of the software system at time of delivery.

3.6.2 Availability

This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

3.6.3 Security

This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure.

3.6.4 Maintainability

This should specify attributes of software that relate to the ease of maintenance of the software itself.

3.6.5 Portability

This should specify attributes of software that relate to the ease of porting the software to other host.

4. Software Design Document

4.1. Structural Design

- a. Class Diagrams
- b. Object Diagrams

4.2. Data Design

- a. Data-flow Diagram

4.3. Behavioural Design

- a. Sequence Diagram
- b. Collaboration diagram
- c. Activity Diagram
- d. State-chart diagram

4.3. Implementation Design

- a. Component Diagram

4.4. Environment Design

- a. Deployment Diagram

5. References

This section should

- 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.

This information may be provided by reference to an appendix or to another document.