

Software Design Specification

1. Introduction

1.1 Purpose of this document

Full description of the main objectives of the SDS document.

1.2 Scope of the development project

This will be similar to what was written in the SRS.

1.3 Definitions, acronyms, and abbreviations

Be sure to alphabetize!

1.4 References

This section will include technical books and documents related to design issues. At a minimum, this should reference the SRS! Give links to documents as appropriate. Be certain that the references you give are complete and in the appropriate format.

1.5 Overview of document

A short description of how the rest of the SDS is organized and what can be found in the rest of the document. This is not simply a table of contents. Briefly describe various parts.

2. System architecture description

This section is the main focus in the first version of the SDS, the high level design. This should give a good view of exact organization of the system as per the requirements.

2.1 Overview of modules / components

This subsection should introduce various components and subsystems.

2.2 Structure and relationships

* Make clear the interrelationships and dependencies among various components. * Structure charts can be useful here. Include explanatory text for better understanding .

2.3 User interface issues

This section will present the main principles of the product's user interface. Use the users defined in user class section of your SRS to make specific examples. This section should not touch on technical details. You may want to include sketches and specific text messages.

3. Detailed description of components

NOTE: This section is the main focus in version 2.0 of the SDS, the detailed design. This section will provide most of the basis for implementing the product.

3.1 Component template description

This section is not part of design. It is the pattern you will use to describe the components given in subsections 3.2 - 3.n. Each part of the template will be identified by a label. Here in 3.1, you must briefly explain the purpose of each point. To make the presentation clear, use a table or bullet list. You may adapt the template suggested below. If you use a Design tool the following is what is recorded in the data dictionary.

Identification	The unique name for the component and the location of the component in the system.
Type	A module, a subprogram, a data file, a control i procedure, a class, etc
Purpose	Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS, but are implied or adjunct to formally stated SDS requirements
Function	What the component does, the transformation process, the specific inputs that are processed, the algorithms that are used, the outputs that are produced, where the data items are stored, and which data items are modified.
Subordinates	The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part.
Dependencies	How the component's function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and data sharing), and responsibility for creation, duplication, use, storage, and elimination of components.
Interfaces	Detailed descriptions of all external and internal interfaces as well as of any mechanisms for communicating through messages, parameters, or common data areas. All error messages and error

	codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here.
Resources	A complete description of all resources (hardware or software) external to the component but required to carry out its functions. Some examples are internet connection, databases, memory, CPU requirements if any, I/O channels, printers, cdwriters, libraries, and system services etc.,
Processing	The full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic.
Data	For the data internal to the component, describe the representation method, initial values, use, semantics, and format.

3.2 X Component (or Class or Function ...)

Use exactly the template you define in 3.2. If a part of the template is not applicable, then mark it N/A rather than omitting it.

3.3 Y Component (or Class or Function ...)

...

3.n Z Component (or Class or Function ...)

4. Reuse and relationships to other products

Reuse can be an important strategy. In some cases, there is freeware that could be incorporated. In other cases, there are existing modules or classes that could be adapted.

5. Design decisions and tradeoffs

Use this section to explain any decisions that can help in better understanding of the design . This section can also capture good ideas that were abandoned and the reasons for leaving them out of the design.

6. Pseudocode for components

7. Appendices (if any)