

SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) TEMPLATE**Version A.4, January 2014****FOREWORD**

This document was written to provide software development projects with a template for generating a System and Software Requirements Specification (SSRS). This document is based on a template originally written by the U.S. Navy Research, Development, Test and Evaluation Division in June 1997 in accordance with the MIL-STD-498 DID (DI-IPSC-81433). The template was updated by the University of Idaho's Center for Secure and Dependable Systems (CSDS) in June 2008 to adhere to IEEE Std. 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*¹, and IEEE Std. 12207-2008, *Systems and Software Engineering – Software Life Cycle Processes*². It was then adapted in September 2008, 2010, 2013, and 2014 for use in UI CS 383.

The SSRS template begins on the next page. Just throw away this page and enter your project specifications into the following template. Don't forget to change the headers and footers as necessary.

DOCUMENT CONVENTIONS

[Text] Replace this text with your project specification text.

text in italics Notes or instructions to the author. **Delete in submitted document.**

¹ IEEE Std. 830-1998, *Recommended Practice for Software Requirements Specifications*, Institute of Electrical and Electronics Engineers, 345 East 47th St. New York, NY, USA, 10017-2394.

²ISO/IEC 12207, IEEE Std. 12207-2008, *Systems and software engineering – Software life cycle processes*, 2nd ed., Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ, USA, 08854.

SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR

[insert your project name]

[replace image above with a cooler logo]



**Version [[insert version number]]
[[insert date]]**

**Prepared for:
[insert company name, address, and contact info]**

**Prepared by:
[insert your name(s)]
University of Idaho
Moscow, ID 83844-1010**

RECORD OF CHANGES

[illegible]

***A** - ADDED **M** - MODIFIED **D** - DELETED

**[INSERT YOUR PROJECT NAME] SSRS
TABLE OF CONTENTS**

Section	Page
1 Introduction	1
1.1 IDENTIFICATION	1
1.2 PURPOSE	1
1.3 SCOPE	1
1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	1
1.5 REFERENCES	2
1.6 OVERVIEW AND RESTRICTIONS	2
2 OVERALL DESCRIPTION	3
2.1 PRODUCT PERSPECTIVE	3
2.2 PRODUCT FUNCTIONS	3
2.3 USER CHARACTERISTICS	3
2.4 CONSTRAINTS	3
2.5 ASSUMPTIONS AND DEPENDENCIES	3
2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS	4
2.6.1 Site dependencies	4
2.6.2 Safety, security and privacy requirements	4
2.6.3 Performance requirements	4
2.6.4 System and software quality	4
2.6.5 Packaging and delivery requirements	4
2.6.6 Personnel-related requirements	5
2.6.7 Training-related requirements	5
2.6.8 Logistics-related requirements	5
2.6.9 Other requirements	5
2.6.10 Precedence and criticality of requirements	5
3 SPECIFIC REQUIREMENTS	6
3.1 EXTERNAL INTERFACE REQUIREMENTS	6
3.1.1 Hardware Interfaces	6
3.1.2 Software Interfaces	6
3.1.3 User Interfaces	6
3.1.4 Other Communication Interfaces	6
3.2 SYSTEM FEATURES	1
3.2.1 Use Case Diagrams	1
3.2.2 System feature 1: [insert feature name here]	1
Non-task feature description	2
Introduction/Purpose	2
Input/Output sequence	2
Design constraints	2
Performance requirements	2
Detailed functional requirements	2
Functional requirement 1.1	2

Functional requirement 1.2	2
.	2
.	2
Functional requirement 1.[n]	2
3.2.3 System feature 2: [insert feature name here]	2
Write another feature description, as either a use case or a non-task feature description	2
. . . do as many as necessary, probably a lot	2
3.2.4 System feature [m]: [insert feature name here]	2
Write a final feature description, as either a use case or a non-task feature description	2
.	2
4 REQUIREMENTS TRACEABILITY	1
5 APPENDIX A. [insert name here]	1
6 APPENDIX B. [insert name here]	1

1 Introduction

This section the document should introduce the project, customer, audience, etc., without delving into too much detail, because those details are provided in subsequent sections.

1.1 IDENTIFICATION

This paragraph shall contain a full identification of the system and the software to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).

The software system being considered for development is referred to as [insert name and or id number]. The customer providing specifications for the system is [insert name and contact info]. The ultimate customer, or end-user, of the system will be [insert name and contact info]. This is a [new | redesign] project effort, so the version under development is version [insert version and release number].

1.2 PURPOSE

This paragraph shall contain a brief statement on the purpose of the system and software being developed, and the intended audience for this document.

The purpose of the system under development is to [insert your text here]. While the system will be used by [insert intended users], this document is intended to be read and understood by UI CS software designers and coders. [Optional: The document will also be vetted or approved by [insert approval people]].

1.3 SCOPE

This paragraph shall briefly summarize the history of system development, operation, and maintenance; identify the project sponsor, acquirer, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.

[insert your text here]

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

This section shall list and define all special terms, acronyms and abbreviations used throughout this document. A tabular form is preferable, but not mandatory.

Term or Acronym	Definition
Alpha test	Limited release(s) to selected, outside testers
Beta test	Limited release(s) to cooperating customers wanting early access to developing systems
Final test	aka, Acceptance test, release of full functionality to customer for approval
DFD	Data Flow Diagram
SDD	Software Design Document, aka SDS, Software Design Specification
SRS	Software Requirements Specification
SSRS	System and Software Requirements Specification

1.5 REFERENCES

This section shall list full bibliographic citations of all documents referenced in this report. This section shall also identify the source for all materials not available in printed form (e.g., web-based information) and list the complete URL along with owner, author, posting date, and date last visited.

[insert your citations here]

1.6 OVERVIEW AND RESTRICTIONS

This paragraph shall describe the organization of this document and shall describe any security or privacy considerations associated with its use.

This document is for limited release only to UI CS personnel working on the project and [state others who will receive the document].

Section 2 of this document describes the system under development from a holistic point of view. Functions, characteristics, constraints, assumptions, dependencies, and overall requirements are defined from the system-level perspective.

Section 3 of this document describes the specific requirements of the system being developed. Interfaces, features, and specific requirements are enumerated and described to a degree sufficient for a knowledgeable designer or coder to begin crafting an architectural solution to the proposed system.

Section 4 provides the requirements traceability information for the project. Each feature of the system is indexed by the SSRS requirement number and linked to its SDD and test references.

Sections 5 and up are appendices including original information and communications used to create this document.

2 OVERALL DESCRIPTION

This section of the document should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides the background for those requirements, which are defined in detail in Section 3.

2.1 PRODUCT PERSPECTIVE

This subsection of the document should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the document defines a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software. A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

[insert your text here]

2.2 PRODUCT FUNCTIONS

This subsection of the document should provide a summary of the major functions that the software will perform. For the sake of clarity, the functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time. Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

[insert your text here]

2.3 USER CHARACTERISTICS

This subsection of the document should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state specific requirements, but rather should provide the reasons why certain specific requirements are later specified in Section 3 of this document.

[insert your text here]

2.4 CONSTRAINTS

This subsection of the document should provide a general description of any other items that will limit the developer's options. These include: a) Regulatory policies; b) Hardware limitations (e.g., signal timing requirements); c) Interfaces to other applications; d) Parallel operation; e) Audit functions; f) Control functions; g) Higher-order language requirements; h) Signal handshake protocols; i) Reliability requirements; j) Criticality of the application; k) Safety and security considerations.

[insert your text here]

2.5 ASSUMPTIONS AND DEPENDENCIES

This subsection of the document should list each of the factors that affect the requirements stated in the document. These factors are not design constraints on the system and/or software but are, rather, any changes to them that can affect the requirements in the document. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the document would then have to change accordingly.

[insert your text here]

2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS

This subsection of the document should identify system level (whole, not functional) requirements that impact the construction, operation, packaging and delivery of the system and software.

2.6.1 Site dependencies

This paragraph shall specify site-dependent operational parameters and needs (such as parameters indicating operation-dependent targeting constants or data recording). The requirements shall include, as applicable, number of each type of equipment, type, size, capacity, and other required characteristics of processors, memory, input/output devices, auxiliary storage, communications/ network equipment, and other required equipment or software that must be used by, or incorporated into, the system. Examples include operating systems, database management systems, communications/network software, utility software, input and equipment simulators, test software, and manufacturing software. The correct nomenclature, version, and documentation references of each such device or software item shall be provided.

[insert your text here]

2.6.2 Safety, security and privacy requirements

This paragraph shall specify the system requirements, if any, concerned with maintaining safety, security and privacy. These requirements shall include, as applicable, the safety, security and privacy environment in which the system must operate, the type and degree of security or privacy to be provided, and the criteria that must be met for safety/security/privacy certification and/or accreditation.

[insert your text here]

2.6.3 Performance requirements

This paragraph should specify both the static and the dynamic numerical performance requirements placed on the software or on human interaction 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. All of these requirements should be stated in measurable terms. For example, "95% of the transactions shall be processed in less than 1msec."

[insert your text here]

2.6.4 System and software quality

This paragraph shall specify the requirements, if any, concerned with hardware and software quality factors identified in the contract. Examples include quantitative requirements regarding the system's functionality (the ability to perform all required functions), reliability (the ability to perform with correct, consistent results), maintainability (the ability to be easily corrected), availability (the ability to be accessed and operated when needed), flexibility (the ability to be easily adapted to changing requirements), portability (the ability to be easily modified for a new environment), reusability (the ability to be used in multiple applications), testability (the ability to be easily and thoroughly tested), usability (the ability to be easily learned and used), and other attributes.

[insert your text here]

2.6.5 Packaging and delivery requirements

This paragraph shall specify the requirements, if any, for packaging, labeling, handling and delivery of the system being developed to the customer.

The executable system and all associated documentation (i.e., SSRS, SDD, code listing, test plan (data and results), and user manual) will be delivered to the customer on CD's and/or via email, as specified by the customer at time of delivery. Although document "drops" will occur throughout the system development process, the final, edited version of the above documents will accompany the final, accepted version of the executable system.

2.6.6 Personnel-related requirements

This paragraph shall specify the system requirements, if any, included to accommodate the number, skill levels, duty cycles, training needs, or other information about the personnel who will use or support the system under development. These requirements shall include, as applicable, considerations for the capabilities and limitations of humans; foreseeable human errors under both normal and extreme conditions; and specific areas where the effects of human error would be particularly serious. Examples include requirements for color and duration of error messages, physical placement of critical indicators or keys, and use of auditory signals.

The system under development has no special personnel-related characteristics.

2.6.7 Training-related requirements

This paragraph shall specify the system requirements, if any, pertaining to training. Examples include training software, tutorials, or help information to be included in the system.

No training materials or expectations are tied to this project other than the limited help screens built into the software and the accompanying user manual.

2.6.8 Logistics-related requirements

This paragraph shall specify the system requirements, if any, concerned with logistics considerations. These considerations may include: system maintenance, software support, system transportation modes, supply-system requirements, impact on existing facilities, and impact on existing equipment.

[Insert a description of the minimum hardware requirements and OS and application software dependencies here]

2.6.9 Other requirements

This paragraph shall specify additional system level requirements, if any, not covered in the previous paragraphs.

[insert your text here]

2.6.10 Precedence and criticality of requirements

This paragraph shall specify, if applicable, the order of precedence, criticality, or assigned weights indicating the relative importance of the requirements in this specification. Examples include identifying those requirements deemed critical to safety, to security, or to privacy for purposes of singling them out for special treatment. If all requirements have equal weight, this paragraph shall so state.

[insert your text here]

3 SPECIFIC REQUIREMENTS

This section of the document 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 into the system, every output from the system, and all functions performed by the system in response to an input or in support of an output. As this is often the largest and most important part of the document, all requirements should be uniquely identifiable and careful attention should be given to organizing the requirements to maximize readability.

3.1 EXTERNAL INTERFACE REQUIREMENTS

This subsection should be a detailed description of all inputs into and outputs from the software system. It should complement the constraints and dependencies defined in earlier sections, but not repeat that information. Hardware, software, user, and other communication interfaces need to be specified. Use the four subsections listed below or the table on the next page, or some combination of both.

3.1.1 Hardware Interfaces

[insert your text here]

3.1.2 Software Interfaces

[insert your text here]

3.1.3 User Interfaces

[insert your text here]

3.1.4 Other Communication Interfaces

[insert your text here]

External Interface Requirements**Hardware Interfaces**

Name	Source/Destination	Description	Type/range	Dependencies

Software Interfaces

Name	Source/Destination	Description	Type/range	Dependencies

User Interfaces

Name	Source/Destination	Description	Type/range	Dependencies

Other Communication Interfaces

Name	Source/Destination	Description	Type/range	Dependencies

3.2 SYSTEM FEATURES

*Functional requirements should define the fundamental actions (i.e., features) that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These requirements are given in the form of **Use Cases** where possible, denoting a concrete use (discrete user-performable task) of the system. Use case diagrams are followed by use case descriptions, followed by any non-task features. Non-task features are generally listed as “shall” statements starting with “The system shall. . .” These include: a) Validity checks on the inputs; b) Exact sequence of operations; c) Responses to abnormal situations, including error detection, handling and recovery; d) Parameter specification and usage; e) Relationship of outputs to inputs, including formulas for input to output conversion.*

It may be appropriate to partition the functional requirements into sub functions or subprocesses, but that decomposition (here) does not imply that the software design will also be partitioned that way. You should repeat subsections 3.2.i for every specified feature defined for the system or software.

3.2.1 Use Case Diagrams

[insert 1+ use case diagrams here]

3.2.2 System feature 1: [insert feature name here]

*For each feature, you should either provide a Use Case Description **or** a Non-task feature description, whichever is more appropriate.*

Use Case Description	Non-task feature description
Name Actors Goals Preconditions Summary Related use cases Steps 1. ... 2. ... Alternatives Postconditions	Introduction/Purpose [insert your text here] Input/Output sequence [insert your text here] Design constraints [insert your text here] Performance requirements [insert your text here] Detailed functional requirements Functional requirement 1.1 [insert your text here] Functional requirement 1.2 [insert your text here] ... Functional requirement 1.[n] [insert your text here]

3.2.3 System feature 2: [insert feature name here]

Write another feature description, as either a use case or a non-task feature description

... do as many as necessary, probably a lot

3.2.4 System feature [m]: [insert feature name here]

Write a final feature description, as either a use case or a non-task feature description

4 REQUIREMENTS TRACEABILITY

This section shall contain traceability information from each system requirement in this specification to the system (or subsystem, if applicable) requirements it addresses. A tabular form is preferred, but not mandatory.

Feature Name	Req No.	Requirement Description	Priority	SDD	Alpha Release		Beta Release	
					Test Case(s)	Test Res.	Test Case(s)	Test Res.
	1.1							
	1.2							
	...							
	1.[n]							
	2.1							
	2.2							
	...							
	2.[n]							
	3.1							
	3.2							
	...							
	3.[n]							
...	...							
	[m].1							
	[m].2							
	...							
	[m.n]							

Priorities are: **M**andatory, **L**ow, **H**igh

SDD link is version and page number or function name.

Test cases and results are file names and **P**ass/**F**ail or % passing.

5 APPENDIX A. [insert name here]

Include copies of specifications, mockups, prototypes, etc. supplied or derived from the customer. Appendices are labeled A, B, . . . n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

6 APPENDIX B. [insert name here]

[insert appendix B here]