**SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) TEMPLATE**

**Version A.1, September 2008**

**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]](#footnote-1), and IEEE Std. 12207-2008, *Systems and Software Engineering – Software Life Cycle Processes[[2]](#footnote-2)*. It was then adapted in September 2008 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 final format.

**SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR**

**Data Map**

[replace image above with a cooler logo]

Version 1.0

12/28/2017

Prepared for:

Northwest Knowledge Network

875 Perimeter Drive, MS 2358

Moscow, ID 83844-2358

Prepared by:

Casey Blair

Northwest Knowledge Network

875 Perimeter Drive, MS 2358

Moscow, ID 83844-2358

**Data Map SSRS**

**RECORD OF CHANGES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Change number | Date completed | Location of change (e.g., page or figure #) | **A M D** | Brief description of change | Approved by (initials) | Date Approved |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

\***A** - ADDED **M** - MODIFIED **D** - DELETED **[insert your project name] SSRS**

**TABLE OF CONTENTS**

**Section Page**

**1.** **1. Introduction** 8

**1.1** **IDENTIFICATION** 8

**1.2** **PURPOSE** 8

**1.3** **SCOPE** 8

**1.4** **DEFINITIONS, ACRONYMS, AND ABBREVIATIONS** 8

**1.5** **REFERENCES** 9

**1.6** **OVERVIEW AND RESTRICTIONS** 9

**2.** **OVERALL DESCRIPTION** 10

**2.1** **PRODUCT PERSPECTIVE** 10

**2.2** **PRODUCT FUNCTIONS** 10

**2.3** **USER CHARACTERISTICS** 11

**2.4** **CONSTRAINTS** 11

**2.5** **ASSUMPTIONS AND DEPENDENCIES** 11

**2.6** **SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS** 12

2.6.1 Site dependencies 12

2.6.2 Safety, security and privacy requirements 12

2.6.3 Performance requirements 12

2.6.4 System and software quality 12

2.6.5 Packaging and delivery requirements 13

2.6.6 Personnel-related requirements 13

2.6.7 Training-related requirements 13

2.6.8 Logistics-related requirements 13

2.6.9 Other requirements 13

2.6.10 Precedence and criticality of requirements 14

**3.** **SPECIFIC REQUIREMENTS** 15

**3.1** **EXTERNAL INTERFACE REQUIREMENTS** 15

3.1.1 Hardware Interfaces 15

3.1.2 Software Interfaces 15

3.1.3 User Interfaces 15

3.1.4 Other Communication Interfaces 15

**3.2** **SYSTEM FEATURES** 17

3.2.1 Use Case Diagrams 17

3.2.2 System feature 1: [ insert feature name here ] 17

Non-task feature description 17

3.2.2.1 Introduction/Purpose of this feature 17

3.2.2.2 Input/Output sequence for this feature 17

3.2.2.3 Design constraints of this feature 17

3.2.2.4 Performance requirements of this feature 17

3.2.2.5 Detailed functional requirements of this feature 17

3.2.2.5.1 Functional requirement 1.1 17

3.2.2.5.2 Functional requirement 1.2 17

3.2.2.6 17

… 17

3.2.2.6.1 Functional requirement 1.[n] 17

3.2.3 System feature 2: [ insert feature name here ] 18

3.2.3.1 Introduction/Purpose of this feature 18

3.2.3.2 Input/Output sequence for this feature 18

3.2.3.3 Design constraints of this feature 18

3.2.3.4 Performance requirements of this feature 18

3.2.3.5 Detailed functional requirements of this feature 18

3.2.3.5.1 Functional requirement 2.1 18

3.2.3.5.2 Functional requirement 2.2 18

… 18

3.2.3.5.3 Functional requirement 2.[n] 18

… 18

3.2.4 System feature [m]: [ insert feature name here ] 18

3.2.4.1 Introduction/Purpose of this feature 18

3.2.4.2 Input/Output sequence for this feature 18

3.2.4.3 Design constraints of this feature 18

3.2.4.4 Performance requirements of this feature 18

3.2.4.5 Detailed functional requirements of this feature 18

3.2.4.5.1 Functional requirement [m].1 18

3.2.4.5.2 Functional requirement [m].2 18

… 18

3.2.4.5.3 Functional requirement [m.n] 18

**4.** **REQUIREMENTS TRACEABILITY** 19

**5.** **APPENDIX A. [insert name here]** 20

**6.** **APPENDIX B. [insert name here]** 21

# **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.*

## **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 being considered for development is referred to as the Data Map. This system is a social network and a social network visualization tool for managing personnel, researchers, publications, data, and oranizations. The current version is version 1, and the current release number is 1.

## **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.*

This system is meant to be a social network for an oranization that a manager can use to see how people and other entities in their organization are connected. Also, it acts as a search system for researchers and employees to find data, publications, people, and institiutions in their organization that they can then contact, download data, or coordinate with.

The system was originally made for the MILES-EPSCOR project to allow new researchers to the system to find researchers working on similar research questions so they could share data. This would allow for a much more efficient use of research grant money as data from similar studies could be re-used in the system. Also, it allows managers of the MILES program to see how the research questions have changed over time as the MILES program progressed to make it easier to report how the MILES program is meeting the Univerity of Idaho’s strategic goals.

## **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.*

The system was developed by the Northwest Knowledge Network. It is currently hosted on the Northwest Knowledge Network’s Docker containers system. It was sponsored by the MILES-EPSCOR program. The intended use is a member of the MILES-EPSCOR project. Currently, the system is hosted at <https://www.idahoecosystems.org/datamap> .

## **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** |
| NKN | Northwest Knowledge Network |
| Docker | Container system used for orchestrating application deployment |
| OS | Operating System |
| DOM | Document Object Model: hierarchical tree structure of HTML elements in website |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## **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.*

Creative Commons Attribution-NonCommercial-ShareAlike 3.0 United States licence: <https://creativecommons.org/licenses/by-nc-sa/3.0/us/>

## **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 anyone using a version of this system. This system is licenced with the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 United States licence. See the text of the license by using the appropriate link in the references section (1.5).

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.**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.*

## **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.*

The Data Map is a social network: it tracks relations in between entities in the network. It does so using lines (relations) in the graph visualization. This social network differs from other social networks, such as Facebook, in that the user can see an overview of all the connections in between all entities in the system. Facebook only allows the authenticated user to see user data from people they search for, or are already following, but only in text format. Facebook does not show a graph visualization of how people are connected. The Data Map does, however.

The Data Map’s database can be populated with any data to make a new social network. The system is open source and can be used by any organization. Some HTML will need to be changed for the filter functionality to work with a new data schema. But other than that, the data schema is read by the JavaScript when the page is first loaded, which makes the amount of code necessary to change for a new data set minimal.

## **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.*

The system will allow managers of an organization to see collaboration between the entities in the system, so they can make management decisions (make two employees work together, merge departments that provide similar services, etc.).

The system will allow non-managers to search for data, publications, and other employees in the MILES-EPSCOR system (or any organization) so they can use already exisiting data, publications, or contact employees to collaborate with. This will be done though the graph visualization system: the user types in a search query, and the graph displays the search results.

A timeline slider will allow managers to see how the organization has changed over time. The slider will filter nodes in the graph by the node’s active years attibute, so the manager can see what projects and people were active in the system at what time.

## **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.

The users of this system are researchers in the MILES-EPSCOR program. They can be undergraduates, graduate students, professors, and MILES-EPSCOR managers. They are scientists in natural sciences.

## **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.

There is a hard deadline for funding for this project: end of May, 2017. Therefore, a working system must be completed before then.

This system will use Docker to help automate deployment to NKN’s new Docker/Rancher system and help manage dependencies for development across multiple platforms.

This system must be available on the internet at the idahoecosystems.org website. Since there is already a Drupal website at idahoecosystems.org, we have to load the Data Map in to a page in the Drupal system using an iframe.

The data schema will have a lot of many-to-many relations. Therefore, a database that can scale well with a lot of many-to-many relations is needed.

There will be FERPA protected data in the MILES-EPSCOR database. Therefore, security is paramount as we must make sure the FERPA data stays confidential.

## **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.

The web server will use Docker. Any operating system that Docker can be installed on (OSX, Windows, Linux, etc.) can host this system.

Dependency management is hendled though the Docker system: specifically, the Dockerfiles included in the Git repository. All dependencies are downloaded though their container’s package management system (eg: apt-get, npm, etc.).

The web server must be connected to the internet or an intranet to be used by others.

## **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.*

### 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 ]

### 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.*

FERPA data is stored in the database. FERPA sensitive data will never be shared to anyone who is not an manager in the organization. Input sanitization of user search text is a priority, and unauthorized users must not be able to access FERPA protected data (any student data).

### Performance requirements

*This paragraph should specify both the static and the dynamic numerical performance requirements placed on the soft ware 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.”*

The system must be able to handle 30 concurrent users. The database will only be updated once a month in a bulk update. Users outside of the Data Map website should not be able to access the data, so a RESTful API is not required.

### 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.

The system must be portable across different operating systems. Since the system is to be hosted on NKN servers, and there are many different server types (RedHat, CentOS, Ubuntu, Windows, etc.) making a system that manages dependecies for us will greatly improve portability if the server needs to be moved to a different OS.

### 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 via a NKN Git repository. This Git repository will be linked to NKN’s Jenkins system for automated deployment to a Docker environment.

### 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.

### 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.

### 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.*

Any operating system that Docker can be installed on is required for this system to work. For example, Docker works on Ubuntu, CentOS, RedHat, Windows 7, Windows 8, and Windows 10.

### Other requirements

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

[ insert your text here ]

### 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 ]**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.*

## **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.*

### Hardware Interfaces

[ insert your text here ]

### Software Interfaces

[ insert your text here ]

### User Interfaces

[ insert your text here ]

### Other Communication Interfaces

[ insert your text here ]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| External Interface Requirements  **Hardware Interfaces** | | | | | |
| **Name** | **Source/Destination** | **Description** | **Type/range** | **Dependencies** | **Formats** | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
| **Software Interfaces** | | | | | |
| **Name** | **Source/Destination** | **Description** | **Type/range** | **Dependencies** | **Formats** | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
| **User Interfaces** | | | | | |
| **Name** | **Source/Destination** | **Description** | **Type/range** | **Dependencies** | **Formats** | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
| **Other Communication Interfaces** | | | | | |
| **Name** | **Source/Destination** | **Description** | **Type/range** | **Dependencies** | **Formats** | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |

## **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.

### Use Case Diagrams

[insert 1+ use case diagrams here]

### System feature 1: [ insert feature name here ]

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

### System feature 2: [ insert feature name here ]

#### Introduction/Purpose of this feature

[ insert your text here ]

#### Input/Output sequence for this feature

[ insert your text here ]

#### Design constraints of this feature

[ insert your text here ]

#### Performance requirements of this feature

[ insert your text here ]

#### Detailed functional requirements of this feature

##### Functional requirement 2.1

[ insert your text here ]

##### Functional requirement 2.2

[ insert your text here ]

##### …

##### Functional requirement 2.[n]

[ insert your text here ]

##### …

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

#### Introduction/Purpose of this feature

[ insert your text here ]

#### Input/Output sequence for this feature

[ insert your text here ]

#### Design constraints of this feature

[ insert your text here ]

#### Performance requirements of this feature

[ insert your text here ]

#### Detailed functional requirements of this feature

##### Functional requirement [m].1

[ insert your text here ]

##### Functional requirement [m].2

[ insert your text here ]

##### …

##### Functional requirement [m.n]

[ insert your text here ]

# **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** | | **Final Test** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case(s)** | **Test Res.** | **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.

# **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 ]

# **APPENDIX B. [insert name here]**

[ insert appendix B here ]

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. [↑](#footnote-ref-1)
2. ISO/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. [↑](#footnote-ref-2)