

CS 448 - Assignment - Requirements Specification Document

Check Canvas (<https://learn.nmsu.edu>) for the Due Dates

0 Guidelines

Your project deliverables must include a requirements specification document. This document describes what your system will do and the environment, constraints, and criteria under which it will perform its duty. Someone unfamiliar with your project or with the application domain should be able to read this document and end up with a strong understanding of what your system needs to do. In fact, they should be able take this document and go off and build their own version of the system and theirs should largely do the same thing yours does – that's how much detail should be in this document!

The rest of this document is an outline, with description, of your required specification document. Your document should begin with the Section 1 that follows and include everything from there on. This outline is adapted from Figure 4.26 in the textbook *Software Engineering: Theory and Practice*, 4th Edition, by Pfleeger and Atlee, Prentice-Hall, 2010; that figure is itself an adaptation of the outline of a Software Requirements Specification defined in an IEEE standard.

1 Introduction

Sections in any document should always have at least some text before a subsection title begins. Here you should provide a short introduction of your project without repeating the material that follows in subsections.

1.1 Purpose of Product

Describe here what the product is accomplishing. In other words, what is the problem that the product will solve? Explain the problem domain in a way that makes it clear there is a need for this product.

1.2 Scope of Product

Describe the extent of what the product will encompass, and most importantly include what it will not encompass. How much of your problem domain will be supported by your product, and what will you leave out?

1.3 Acronyms, Abbreviations, Definitions

Explain any vocabulary and acronyms that are used in the problem domain that might not be readily understandable to a general software person who is not active in that domain.

1.4 References

List any documents (books, manuals, papers, web sites, etc.) that you are using as information about your problem domain and the environment your product will run in. If your users are requiring a specific platform that you develop on, then include documents about this, but if your platform is a design choice, leave it for the design specification.

2 General Description of Product

This section should provide an overview of the product and its environments. Section 3 will delve into the details.

2.1 Domain Model

This section must include a UML class diagram that captures the problem domain information structure in the form of the entities (classes), their attributes, and their relationships to one another. In addition this section must include explanatory text that describes the domain diagram and elucidate in particular any important points that occur in the model. This model is only about the information structure in the problem domain, not about any solution design or any classes in an eventual product.

2.2 Context of Product

Describe the environment the product will be operating in. What other systems will it be interacting with? What type of computing environment? What type of operating environment? (office? home? warehouse? outdoors?)

2.3 Product Functions

Describe the general capabilities that the product will provide to users (or other components if it is an embedded system). Think of this section as a short overview of Section 3.2. This section is of dubious value but appears in many suggested specification outlines. It would be OK to leave it out.

2.4 User Characteristics

Describe your expected user. What is their level of skill? What are your assumptions about their abilities? What is the minimum ability level you expect? If there are different user roles, do this for each role.

2.5 Constraints

What external constraints are there on how you design your system? Is the platform (computer, operating system, etc.) chosen for you or do you get to choose? Do you have to adhere to standards? Which ones? (e.g., http, ssl, specific domain file formats, etc.)

2.6 Assumptions and Dependencies

Describe any other assumptions that have not been described earlier, and dependencies that you have. Must you integrate with other software? hardware?

3 Specific Requirements

This is the detailed heart of the requirements specification. Here is where you really need

to get down to the specific details of what your system must do. This should be the bulk of your document.

3.1 External Interface Requirements

This is where you describe in detail the external interactions that your system must have. If your system is user-oriented, then probably the first subsection will be the most prominent, and the other three might be able to be collapsed into one "Other Interfaces" section. You can refer to external documents in these sections where needed; e.g., if your system is web-based you could refer to the most recent HTTP protocol specification and HTML specification (will you use HTML4 or HTML5?).

3.1.1 User Interfaces

Remember that this is a requirements document, not a design document. Do not make the mistake of designing the interfaces here, but only in describing what specific interfaces you will need to support the required functionality.

3.1.2 Hardware Interfaces

If your system must interface to hardware, describe it here.

3.1.3 Software Interfaces

If your system will directly interact with other software packages, describe that here. E.g., if you are building a browser plugin, your system will need to adhere to the existing plugin API.

3.1.4 Communications Interfaces

If your system will be performing communication with other entities, describe the protocol requirements here. (Refer to external specifications where appropriate).

3.2 Functional Requirements

This subsection is the heart of the specific requirements. You should make a very detailed list of the specific behaviors that the software will perform as part of performing the entire problem solution. Be as detailed as possible. You should partition the behaviors into various subsets, and detail each set in a subsection. For example, if the software has various modes, the behavior in each mode can be in a subsection (e.g., maybe a game a setup mode, various play modes (normal, power-enhanced, etc.), a map mode, etc.; each of these could be a subsection. The subsections should be named appropriately, not "Set One", "Set Two", and so on.

Specific complex behaviors that involve multiple steps from users or even multiple users would be useful to document using UML activity diagrams.

3.2.1 Set One

3.2.2 Set Two

3.2.3 Set Three

...

3.3 Performance Requirements

Describe here the detailed performance requirements of your system. Be as concrete as

possible: e.g., "fast" is not concrete, but "at least 15 frames per second" is.

3.4 Design Constraints

Describe here any constraints that force you to make specific design decisions. Maybe the customer wants a Model-View-Controller architecture? Maybe you must use a programming framework that enforces a particular architecture.

3.5 Quality Requirements

What is the user going to expect in terms of quality? If this is an embedded system, what is the mean time to failure that is allowed? How will it be reset? What failure rate is acceptable to the user? What data is allowed to be lost, and what data must never be lost. How will system recovery happen?

3.6 Other Requirements

Anything else that is important but did not fit in other sections.

4 Appendices

Detailed information that was not appropriate to put in-line in any section. Perhaps a protocol specification or particular file layout that must be adhered to. Domain-specific detail can be here to, such as physics equations used in a simulation, or a detailed description of some device that needs to be used, or some particular accounting method that must be followed. If you have developed use cases as part of your requirements gathering, you could put those in an appendix.