Project Deliverable 2 (Requirements Document)

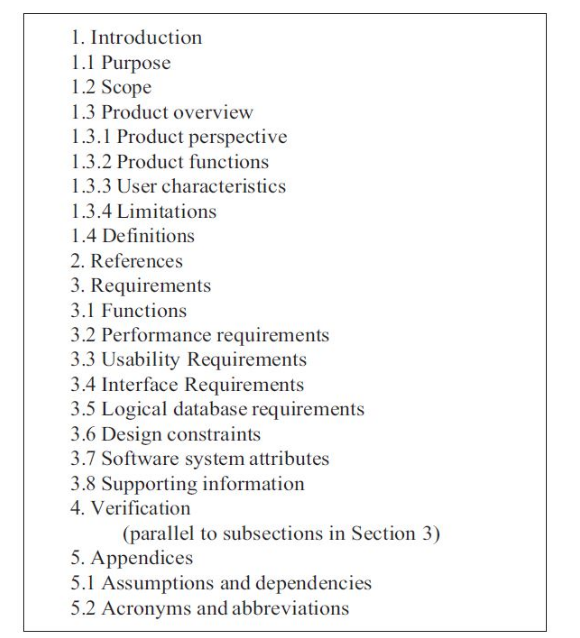


CS-UH

2012

Software Engineering

This document provides a description of the Software Requirements Specification (SRS) template you will use in the second project deliverable (requirements document). **This deliverable is marked out of 80 points. You must submit it as a PDF on Brightspace.**



# 1. Introduction (Total 21.5 points)

1.1 **Purpose (1 point)**: Define the reason(s) for which the system is being developed or modified

1.2 **Scope (3 points)**: Define the scope of the system under consideration by:  
a) identifying the system to be produced by name;  
b) referring to and stating the results of the earlier finalized needs analysis, in the form of a brief but clear expression of the user's problem(s). It explains what the system will and will not do to satisfy those needs;  
c) describing the application of the system being specified. As a portion of this, it should describe all relevant top-level benefits, objectives, and goals as precisely as possible.

1.3.1 **Product Perspective (4 points)**: Define the system's relationship to other related products.  
If the product is an element of a larger system, relate the requirements of that larger system to the functionality of the product covered by the SRS.  
If the product is an element of a larger system, identify the interfaces between the product covered by the SRS and the larger system of which the product is an element.  
Consider a block diagram showing the major elements of the larger system, interconnections, and external interfaces. You can express this using a context diagram (introduced in the Requirements lecture).

1.3.2 **Product Functions (5 points)**: Provide a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statement and invoice preparation without mentioning the vast amount of detail that each of those functions  
requires**. You must have at least 5 major functions.**

1.3.3 **User Characteristics (2 points)**: Describe the general characteristics of the intended groups of users of the product including characteristics that may influence usability, such as educational level, experience, disabilities, and technical expertise. This description should not state specific requirements, but rather should state the reasons why certain specific requirements are later specified.

1.3.4 **Limitations (4.5 points)**: Provide a general description of any other items that will limit the supplier's options, including the following:   
a) regulatory requirements and 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 (e.g., XON-XOFF, ACK-NACK);  
i) quality requirements (e.g., reliability);  
j) criticality of the application;  
k) safety and security considerations;  
l) physical/mental considerations; and  
m) limitations that are sourced from other systems, including real-time requirements from the controlled system through interfaces.

1.4 **Definitions (2 points):** Any definitions that the reader needs to be aware of to understand this document.

# 2. References (Total 3.5 points)

# 3. Requirements (Total 46 points)

3.1 **Functions (20 points)**: Define the fundamental actions that must take place in the software/system in accepting and processing the inputs and in processing and generating the outputs. It may be appropriate to partition the functional requirements into sub-functions or sub-processes, according to a certain grouping that makes sense (e.g., user types, system mode, various high-level features etc.) This does not imply that the software/system design will also be partitioned that way. **You must express requirements using shall statements. You must include at least 15 functional requirements.**

3.2 **Performance Requirements (5 points)**:  Define static and dynamic requirements placed on the software/system or on human interaction with the software / system. This might include:

* the number of simultaneous users to be supported
* the numbers of transactions and tasks the amount of data to be processed within certain time periods for both normal and peak workload conditions

3.3. **Usability Requirements (5 points)**: define the requirements measuring the effectiveness in using the system’s functionalities, efficiency, satisfaction criteria, and avoidance of harm that could arise from use in specific contexts of use. These kinds of requirements are hard to discover in novel systems or for new features. Prototyping can be helpful in discovering such requirements. **Accordingly, you are required to provide screenshots corresponding to at least two UI prototypes (whether paper sketches, wireframes, or interactive UI prototypes), corresponding to two major features of your system.**

3.4 **Interface Requirements (5 points)**: Interface requirements define all inputs into and outputs from the system. Each defined interface should include the following content:  
◾ Name of item  
◾ Description of purpose  
◾ Source of input or destination of output  
◾ Valid range, accuracy, and/or tolerance  
◾ Units of measure  
◾ Timing  
◾ Relationships to other inputs/outputs  
◾ Data formats  
◾ Command Formats  
◾ Data items or information included in the input and output

3.5 **Logical Database Requirements (5 points):**  Define Types of information used by various functions such as

* Frequency of use
* Accessing capabilities
* Data entities and their relationships
* Integrity constraints
* Data retention requirements

3.6 **Design Constraints (3 points):** Define requirements related to standards compliance and hardware limitations.

3.7 **Software System attributes (10 points):** software or system attributes often can include reliability, availability, security, maintainability, portability, and many others. In general, any such attribute is termed an “ility” because most end with “ility.” However, other attributes, such as safety, timeliness, security, and others, are also ilities.

3.8 **Additional Information (3 points)**: Additional supporting information that can be considered includes:  
a) sample input/output formats, descriptions of cost analysis studies or results of user surveys;  
b) supporting or background information that can help the readers of the SRS;  
c) a description of the problems to be solved by the software; and  
d) special packaging instructions for the code and the media to meet security, export, initial loading or other requirements.

# 4. Verification (5 points)

Briefly provide the verification approaches and methods planned to ensure the correctness of the software/system. Note: In the first version of this document, you can provide a brief description of the planned verification techniques (as you see fit and based on your own research on verification techniques). In the second version of this document, we expect you to further refine this section with the knowledge you gained about verification techniques in the course.

# 5. Appendices (Total 4 points)

5.1 **Assumptions and dependencies (2 points)**: List each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software/system; however, any changes to these factors can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product.

5.2 **Acronyms and abbreviations (2 points)** : Spell out or define all acronyms and abbreviations used in the documents.