

**CS 460W – Software Development
Spring 2022
Project Deliverable #1**

**Software Requirement Specification (SRS) Document
Due Date: February 10**

In this assignment, you are to prepare a draft software requirement specification document that follows the **Recommended Practice for Software Requirements specifications** document included below. The IEEE standard 830 is a recommended practice for writing software specification requirements. An updated version of these standards available at 'Course Documents' on Blackboard. Your SRS document should include the following requirements: functional requirements, external interfaces, performance, quality attributes and design constraints. Your team is expected to present the draft document to the team's client. Following the presentation, any needed revisions must be implemented and the revised document must be submitted by the above due date. A copy of your SRS document should also be available on the team's web page.

What to submit?

Each team member must submit via Blackboard the SRS document (which should include the team name, all team member names and a link to the teams' webpage where project documents are available), minutes of meetings, as well as a 1-2 page document that describes in detail the team member's individual contribution to the document.

Grading Rubric:

Each team member's grade will be the average of the team grade and the individual grade.

Team Grade:

Correctness and completeness of the requirements: 70%

Minutes of team meetings for this deliverable. Please review the required items to be included in the minutes as described in the project description: 30%

Individual Grade:

Correctness and completeness of your particular contributions as detailed in action items listed in the minutes. Please note that each team member must contribute about equally. 70%

Participation at every team meeting 30%

Recommended Practice for Software Requirements Specifications (Based on IEEE Standard 830)

1. **Introduction:** This section contains an overview of the complete document.
 - 1.1 **Purpose** What is the purpose of this document, and for whom is it written?
 - 1.2 **Scope** An identification of the product to be developed, what does it do (and what does it not do), why is the product being developed (including a precise description of its benefits, goals and objectives)?
 - 1.3 **Definitions, acronyms and abbreviations** This subsection contains definitions of all the terms, acronyms and abbreviations used in the document. Special attention should be paid to the clarification of terms and concepts from the domain of application.
 - 1.4 **References** References to all documents that are referred to in the remainder of the requirements specification.
 - 1.5 **Overview** This subsection contains an outline of the remainder of the document.
2. **General Description** This section contains a description of matters that concern the overall product and its requirements. It provides a perspective for understanding the specific requirements from section 3 of this document.
 - 2.1 **Product perspective** Does it concern an independent product or is it part of a larger product? In the later case, the other components should be identified, and the interfaces with those components should be described. In this section, we also give an identification of the hardware to be used.
 - 2.2 **Product functions** An overview of the functions of the system to be delivered. This should be confined to an overview. A detailed discussion of the functions is given in section 3 of the requirements specification.
 - 2.3 **User characteristics** An indication of general user characteristics, in as far as these are relevant for the requirements specification. Experience, training and technical expertise of future users may influence specific requirements of the system to be developed.
 - 2.4 **General Constraints** An indication of any other constraints that apply. These may concern government regulations, hardware constraints, security regulations, and so on. Again, we are concerned with the rationale at this point. A further elaboration follows in section 3 of this document.
 - 2.5 **Assumptions and dependencies** This does not concern constraints on the system to be developed, but things which may influence the requirements specification once

they change. As an example, we may think of the availability of certain supporting software, such as some given operating system or numeric library. If that operating system or library turns out not to be available, the requirements specification will have to be adapted accordingly.

3. **Specific Requirements** This section contains all the details which are relevant for the design phase to follow. The ordering given here is just one way to present the specific requirements in a logical way. Specific requirements should be such that one may objectively determine whether they are fulfilled or not.

3.1 **Functional Requirements** In this subsection, a description is given of how the transportation of inputs to outputs is achieved. The description given for each class of functions, and sometimes for each individual function. To a certain extent, this description can be seen as a solution to the user. This component of the requirement specification is the main starting point for the design phase.

3.1.1 Functional requirement 1

3.1.1.1 **Introduction** A description of the purpose of this function and the approaches and techniques used. The introduction should include information to clarify the intent of the function.

3.1.1.2 **Inputs** A precise description of the function's inputs (source, quantities, range of acceptable values, and the like).

3.1.1.3 **Processing** A definition of the operations that must be performed, such as checking for acceptable values, reaction to abnormal situations, or a description of algorithms to be used. As an example of the later, one may think of the use of some mathematical model for strength computations within a CAD-program.

3.1.1.4 **Outputs** A precise description of the outputs (destination, quantities, error messages, and the like).

3.2 External interface requirements

3.2.1 **User interfaces** A description of the characteristics of the user interfaces, such as screen layout, function keys, help functions. In order to support testing, verifiable requirements regarding learning time for the system functions should be included either here or in some subsection of 3.5 (Attributes).

3.2.2 **Hardware interfaces** A description of the logical characteristics of hardware interfaces, such as interface protocols, or screen-oriented versus line-oriented terminal control.

3.2.3 **Software interfaces** A description of software needed, such as a certain operating system or subroutine package. Interfaces to other application software is also discussed here.

3.2.4 **Communications interfaces** An example is a communication protocol for LANs.

3.3 **Performance requirements** Performance requirements encompass both static and dynamic requirements. Static requirements concern, amongst others, the number of terminals to be connected and the number of users that can be handled concurrently. Dynamic requirements concern the operational performance of the system: how frequently will certain functions be called for and how fast should the system's reaction be. It is important that these requirements be stated in measurable terms.

3.4 **Design constraints** Design constraints may result from such things as the prescribed use of certain standards or hardware.

3.4.1 **Standards compliance** Which existing standards or regulations must be followed, and what requirements result from these. For example, certain report formats or audit procedures may be prescribed.

3.4.2 **Hardware limitations** A description of the characteristics of the hardware environment, in as far as they lead to software requirements. An example of this might be the amount of memory available.

3.5 **Attributes** In this section, particular attention is paid to quality aspects. These requirements must be measurable and verifiable. They must be stated in objective terms. The subsections below by no means comprise a complete list of such attributes.

3.5.1 **Availability** Factors that guarantee a certain level of availability, such as restart procedures. In this subsection we may also enlist requirements regarding fault tolerance (with respect to both hardware failures and software failures).

3.5.2 **Security Requirements** regarding unauthorized access and other forms of misuse. Certain cryptographic techniques may be prescribed, and we may put constraints on the communication between different parts of the system.

3.5.3 **Maintainability** Requirements to guarantee a certain level of maintainability of the system, such as a maximum allowable coupling between components.

3.6 Other requirements A description of requirements that are specific to certain software, and which have not been discussed yet.