SOFTWARE REQUIREMENTS SPECIFICATION (SRS)  
TEMPLATE FOR CS 2450

This Software Requirements Specification (SRS) template is written in general conformance with **IEEE Standard 830-1984**, **IEEE Guide to Software Requirements Specification.** Some portions of the outline were tailored to the needs of academic projects.

**Purpose and Guidance**

The purpose of this template is to provide you with a model to create a Software Requirements Specifications (SRS) document. The SRS should contain a complete description of the external behavior of the software system to be developed. This is one of several documents that are used to describe the preparation of software-oriented technical documentation, produced as a part of a defined system development life cycle.

It is not the intent of this template to describe or prescribe a particular process or life cycle. The concept and content of the SRS described in this template are compatible with several different product development strategies, from highly research oriented, exploratory projects, to sustaining software maintenance efforts.

The template that follows is intended to provide insight into the information that should be presented in a typical SRS. It is critical to quality of the completed software project that format, representation techniques, and change control be properly planned and implemented before the SRS is started. Considerations must be given to the type of system to be developed (real time or non-real time), the available development tools, and the available resources.

**References**

**Government, National, and International Standards**

ANSI/IEEE Std 830, IEEE Guide to Software Requirements Specifications, Institute of

Electrical and Electronics Engineers, New York, 1994.This standard provides guidance on the preparation and content of Software Requirements Specifications.

DELETE THIS PAGE BEFORE SUBMITTING!

Software Requirements Specification

for

PROJECT

Version X.X

Prepared by

DATE

Instructor Note: Fill out the title page with a project name, a version number, either your name or the name of your ‘development company’, and the date of creation.

Table of Contents

[1. Introduction 4](#_Toc536792501)

[1.1 Purpose and Intended Audience 4](#_Toc536792502)

[1.2 Project Scope 5](#_Toc536792503)

[1.3 Terms, Definitions, and Acronyms 5](#_Toc536792504)

[1.4 References 5](#_Toc536792505)

[2. Overall Description 6](#_Toc536792506)

[2.1 Product Perspective 6](#_Toc536792507)

[2.2 Product Features 6](#_Toc536792508)

[2.3 User Classes and Characteristics 7](#_Toc536792509)

[2.4 Operating Environment 7](#_Toc536792510)

[2.5 Design and Implementation Constraints 7](#_Toc536792511)

[2.6 Assumptions and Dependencies 7](#_Toc536792512)

[3. System Features 8](#_Toc536792513)

[4. Non-Functional Requirements 8](#_Toc536792514)

[5. External Interface Requirements 9](#_Toc536792515)

[5.1 User Interfaces 9](#_Toc536792516)

[5.2 Hardware Interfaces 9](#_Toc536792517)

[5.3 Software Interfaces 9](#_Toc536792518)

[5.4 Communication Interfaces 9](#_Toc536792519)

[6. Detailed Use Cases 9](#_Toc536792520)

[7. Appendix 10](#_Toc536792521)

Instructor Note: Make sure the page numbering is correct especially as you add more content. See the link in the assignment instructions for details on how to update the table of contents.

The link in the assignment instructions includes info on how to add new sections so they will be in the table of contents when you update it.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

Instructor Note: first entry is the creation date. Add new entries for each homework submission

**Instructor Tips:**

* First, complete the 'Lard Factor" discussion assignment to help with wordiness.
* Don’t write this in a possessive tone. Instead of saying “I will” or “we will”, say 'the system will...'
* Don’t write in a passive tone. Instead of saying “allow for”, “might”, or “should”, say 'the system will...'
* Don't use words like "you" or "your". This document is written for more than just the client.
* Don’t use vague or exaggerated words. What exactly does ‘huge’, ‘many’, or ‘all’ mean? The client will decide and hold you to it.
* Don’t use flowery words like "21st century", even if you are mirroring the client’s words. Again, the client will decide what this means and then hold you to it!
* Assume a non-techy reader. If you want to use techy terms like ‘client facing’, ‘online presence’, or ‘internet storefront’ put them in the Terms and Definition section.
* Do not add random features at will. Consult with the client about new / amazing / spectacular functionality you think would benefit them. They might think it’s awesome too and want it added.

# Introduction

The introduction contains a brief summary and description of the product to be developed and its intended use. The purpose of this section is to give the reader a preview of the document's contents and the product to which it applies. A single paragraph should suffice.

Instructor Note: For the Introduction, I would recommend one or two brief paragraphs introducing the client’s business and the problems he is having (the ones your new system will solve) ... and how the proposed system will solve those problems.

## Purpose and Intended Audience

As you prepare the SRS, remember that it is being written for the customer and possibly the end user. It must be understandable to them. The SRS is the contract between you and the customer. It defines which requirements the project will fulfill.

The purpose of the Software Requirements Specification is to provide agreement on what the software product or the portion of the product to be implemented in software is to do. Specifically, it provides a means of:

• communications among the project sponsors, system analysts, software designers, engineers, and software quality engineers,

• supporting software design and software quality assurance activities,

• supporting system testing activities,

• supporting Verification and Validation activities, and

• supporting a controlled system evolution

Instructor Note: This is the purpose of this document, not the purpose of the project. This is some very good verbiage

## Project Scope

If this is part of a larger system, or will only include certain features, then note it. Example: This product will address the student record functionality only. It will not store information on professors or courses offered at the University.

Instructor Note: The scope focuses on the limitations of the system in the area of the business. What is does and does not do. Include what it will NOT do, otherwise the client will hold you to things you may not have intended

This should be a BRIEF one-paragraph summary of the scope: Something like “The product will do X, Y and Z but it will not do A or B”

## Terms, Definitions, and Acronyms

These are not only business terms that your client uses, but also technical (IT) terms that you will be using for the developers. The idea is to make sure both the customer and the developers are on the same sheet, regarding terminology, when they read this document. Don’t assume!

Instructor Note: Create a list of terms: alphabetize the list and bold the terms. Assume a non-techy reader so put techy terms like server and database in this list. If in doubt, put it in here. You do not have to define commonly recognized terms like computer or Customer. And no, most people don’t know what a server is…

Then make sure you are consistent with the name of the term every time you reference it in the document

## References

Provide a list of the documents that directly relate to the preparation and understanding of the product described in this requirements document. Examples include books, articles, and correspondence with the customer or documents describing the current system or operation. In general, any documents provided by the customer or intended user that help define the ideas presented in the requirements should be referenced. Only include documents that were used to develop the technical content of the SRS. Do not include the document guidelines for the course, or reference books on software engineering, etc. For any reference cited, include complete bibliographic information.

Instructor Note: this section lists where you got the information about the **proposed** system and its features? Interviews? Discussions? Reading client documents?

If you have documentation (like the Client Requirement), give the name of the documents here, and indicate that they are in the Appendix. Then make sure they're actually IN the Appendix at the bottom of the document.

Include the System Request and the Cost Benefit Analysis you created in Homework 2. This contains valuable information about the proposed system.

# Overall Description

## Product Perspective

This section should put the product you are developing into perspective with other related activities, products or projects within the customer’s environment. You need to state if the product is independent and totally self-contained, if it must interface with another system (manual or automated), or if it receives data from another system.

If you are developing a product that is a component of a larger system or project, then this section should:

• briefly describe the functions of each existing component of the larger system or project, and identify the interface among the functions.

• identify the principal external interfaces of the new software product.

• provide a block diagram showing the major components of the larger system or project, and the interconnections and external interfaces with your product.

• clearly indicate where and how the user will interact with your new product.

For a product that is stand-alone (not part of a larger system), this section should:

• identify the principal external interfaces of the new software product.

• provide a block diagram that shows the relationship of your product to the

"usage environment."

• clearly indicate where and how the user will interact with your new product.

Instructor Note: The Product Perspective can be one or two paragraphs of high level marketing about the product; like you would see on the software box at BestBuy. Assume the readers have NO technical knowledge. If using techy terms, make sure they are in the Terms and Definition section.

Payment

Processing

## Product Features

You should provide a summary of the features that the product will possess. As an example, for an accounting program the features might include customer account maintenance, preparation of customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires. Features should generally correspond to major functional capabilities of your product and might be something that you would want to include in advertising literature if you were going to market the product commercially. The features should be organized in a way that makes them understandable to the customer or to anyone else reading the document for the first time. Features should be identified and described in terms common to the user's or problem's domain. They are **not** to be confused with specifying the structure of individual portions of programs or subprograms.

Instructor Note: This is also high level marketing. A bullet list of features you might see on the same box at BestBuy. Having a bulleted list feature will be VERY helpful for future additions to the SRS.

## User Classes and Characteristics

This section describes (in general terms) the computer experience and knowledge of the potential user. This section provides the foundation for the performance and usability requirements addressed within Section 3.

Instructor Note: list who will be using the system and their expected skill level with technology, NOT their functionality within the system. There are 6 user groups described in the client document. Break out by user groups since each is expected to have a different set of skills.

Also, the client has named the user groups. Use the names given by the client and don’t make up other users. It’s hard enough to keep track of these six. If you think the system needs other users like an IT Support Technician, put that in the Cost Benefit Analysis and propose a fee schedule.

## Operating Environment

What kind of environment (single/multiple locations, retail, academic, outdoor, etc.) is the system intended to be operated in?

Instructor Note: how, where, under what conditions the system would be used.

## Design and Implementation Constraints

This section details all constraints upon the product being developed, whether customer mandated or policy/regulatory requirements, e.g., a particular operating system.

Instructor Note: limitations on the requirements and/or operating conditions or design requirements that must be satisfied or met.

## Assumptions and Dependencies

This section details all assumptions and dependencies required for the successful completion of a project, e.g., a certain algorithm being available by a certain date.

Instructor Note: things that you expect to happen or things outside of your control that are assumed to be true.

# System Features

The following describes the functional requirements from the client’s perspective and need.

Example: The completed system will provide the following functionality to all users (in general) and will include the ability to:

* Manage (add, update) reservations
* Manage (add, update, delete) user profiles

Specific functionality for managers & owners (only) will include the ability to:

* Manage (add, update, and delete) inventory items
* Override sales totals and offer a discount
* Receive alerts via text messaging when inventory stock is low
* Perform all the functions listed under the employee (legal, maintenance, and clerk) roles

In addition, specific functionality for maintenance personnel will include the ability to:

* Generate vehicle history reports

**VERY IMPORTANT Instructor Note:** Describe the specific functionality of each of the 6 user groups from the client’s prospective, broken out by user group. Think about each user group and the things they should be able to do.

Start each user group with the major functionalities they are responsible for. List each major functionality with the template wording of Manage XXX(add, update, delete) replacing XXX with the functionality they should be able to... manage. Then you can list other minor functionalities. Refer to the ‘managers & owners’ user group for an example of wording for major and minor functionalities.

This format will be VERY helpful in the next homework assignment.

For instance: among other things, Customers should be able to Manage(add, update, delete) Reservations, Manage(add, update, delete) their Profile and Browse the online catalog

# Non-Functional Requirements

* Here you will Address FURPS+ (Group by Category)

Instructor Note: This section is either FURPS+:

* **Functionality** -
* **Usability**
* **Reliability**
* **Performance**
* **Supportability**
* **+ Miscellaneous**

OR you can use the 4 non-functional requirements shown on p.90 of the textbook:

* **Operational** -
* **Performance**
* **Security**
* **Cultural**/**Political**

# External Interface Requirements

## User Interfaces

This section describes any external system user interfaces that may be required by the system, e.g., a console window.

## Hardware Interfaces

This section describes any external system hardware interfaces, e.g., an analog to digital converter that may be required by the system.

## Software Interfaces

This section describes any external software interfaces, e.g., input files to this system that were created by another software system.

## Communication Interfaces

This section describes any external communication interfaces, e.g., TCP/IP communication sockets that are necessary for the operation of this system.

# Detailed Use Cases

Insert UML Diagrams and USE CASE DETAILS here

Instructor Note: In future homework assignments, you will be creating a variety of diagrams, images, and spreadsheets and inserting them into this SRS document. That means submitting ONE document. Make sure to format the artifacts for optimal viewing. Preferably full screen with labels and casual descriptions for each one. Change the orientation of the page <not the entire document> if necessary to increase the size.

# Appendix

Include in the appendix all supporting documentation that is unique to the project and is not accessible in other forms or in other places within the SRS. This may include correspondence concerning the project and materials provided by the customer. Do not include material in the appendix, such as illustrations and diagrams that should be an integral part of a required section of the SRS.

Instructor Note: The appendix is always the **last thing** in the document. As you add more sections in future assignments, put them **above** the appendix. The link in the assignment instructions includes info on how to add new sections so they will be in the table of contents when you update it.

Include the entire contents of all the documents mentioned in the References section, not just a link to the document. Hard copy readers can’t follow a link and if the document is stored on your computer or in Canvas, the reader can’t access it anyway.

There are details in the Assignment instructions on how to “Update the Table of Contents and copy .pdf files in the SRS”. That includes adding images, .pdf files and other objects into the SRS document.

**Delete the first page <the one before the title page>, any unused language from the original template <for example, the instructions under Appendix>, and all Instructor Notes before submitting.**