

# **Software Requirements Specification (SRS)**

## **Project X**

**Team:**

**Authors:**

**Customer:**

**Instructor:**

# 1 Introduction

- Provide an overview of the entire SRS subsections
- Indicate the topics that will be covered in this document.

Start of your text.

## 1.1 Purpose

- What's the purpose of the SRS document?
- Specify the intended audience.

Start of your text.

## 1.2 Scope

- Identify SW product(s) to be produced by name
- Describe the application of SW being specified, including benefits, objectives, goals. What is the application domain? (e.g., embedded system for automotive systems, graphical modeling utility) This is the domain description of the application.
- Explain what SW product will, and if necessary, will not do. This is the requirement of the application.
- Be consistent with similar statements in higher-level specifications (e.g., the original project specification from customer)

Start of your text.

## 1.3 Definitions, acronyms, and abbreviations

- Define all terms, acronyms, and abbreviations need to understand the SRS. If this section is extensive, then move to an appendix. It is also possible to provide a link to other resources for extensive terminology explanation.

Start of your text.

## 1.4 Organization

- Describe what the rest of the SRS contains
- Give the organizational structure of the SRS.

Start of your text.

## 2 Overall Description

- Give a brief introduction of what information will be covered in this section.

Start of your text.

## 2.1 Product Perspective

- Describe the context for the product
- Is it one element that is part of a bigger system? If so, then give a pictorial representation or diagram (e.g., data flow diagram – DFD, block diagram) that describes how your product fits.
- Interface Constraints:
  - System interfaces
  - User interfaces
  - HW interfaces
  - SW interfaces
  - Communication interfaces
- Other types of constraints:
  - Memory
  - Operations
  - Site adaptation operations (customization that is done on-site).

Start of your text.

## 2.2 Product Functions

- Summarize the major functions that software will perform (portions may come directly from the customer specification – cite as appropriate).
- These function descriptions should be easily understandable by the customer or to any general reader.
- Diagrams: (for all diagrams, introduce the notation first)
- Give and describe a high-level goal diagram for system.

Start of your text.

## 2.3 User Characteristics

- Expectations about the user (e.g., background, skill level, general expertise)

Start of your text.

## 2.4 Constraints

- See list of possible constraints from IEEE SRS document.
- Give English descriptions of safety-critical properties
- Give English descriptions of other properties that if violated, the system will not perform properly.

Start of your text.

## 2.5 Assumptions and Dependencies

- Assumptions made about the HW, SW, environment, user interactions.

Start of your text.



## 2.6 Apportioning of Requirements

- Based on negotiations with customers, requirements that are determined to be beyond the scope of the current project and may be addressed in future versions/releases.

Start of your text.

### 3 Specific Requirements

- Give an enumerated list of requirements.
  - As appropriate, use a hierarchical numbering scheme.
1. Sample requirement at the top level
    - 1.1. Level 2 requirement example
    - 1.2. Another Level 2 requirement
  2. Select the “Requirement” Style.

## 4 Modeling Requirements

- This is the specification portion of the requirements document. (Specifying the bridge between the application domain and the machine domain.)
- For each new diagram type introduced, describe the notation.
- Give and describe use case diagrams
- Use the template below to describe each use case.
  - Each goal may be satisfied by 1 or more use cases
  - Each use case should refer to 1 or more requirements (in Section 3)
- Give and describe a high-level class diagram that depicts the key elements of the system
  - Include a data dictionary to describe each class, its attributes, its operations, and relationships between classes.
- Representative Scenarios of System:
- Give English descriptions of representative scenarios for each use cases.
  - Check: use instances of the class names from class diagram; refer to the terms used in use case diagram
- For each scenario, give a corresponding sequence diagram
  - Check: Objects should be instances of classes in class diagram
- Create and explain a state diagram for all key classes that participate in the scenarios (from above).
  - Check: that all scenarios can be validated against the state diagrams.
  - Check that the events, actions are modeled in the class diagram.
  - Check that all variables referenced in the diagrams are declared as attributes in the class diagram.

Start of your text.

Use Case Name:	
Actors:	
Description:	
Type:	
Includes:	
Extends:	
Cross-refs:	
Uses cases:	

## 5 Prototype

- Describe what your prototype will show in terms of system functionality.

## 5.1 How to Run Prototype

- Describe what is needed to run your prototype
  - What system configuration? (Should be accessible through web.) Are there plugins? Are there any OS or networking constraints. Give the URL for the prototype.
  - 
  - Prototype v1 does not have to be executable per se. But there should be sufficient number of interfaces for the customer to understand the development's interpretation of the requirements.
  -
- Prototype V2 should also be accessible via a webpage. It should be executable and provide an interactive interface.

## 5.2 Sample Scenarios

- Give a sample scenario of using your system. Use real data and problem scenarios. Include screen captures illustrating what your prototype produces. As always, be sure to describe all figures.

## 6 References

- Provide list of all documents referenced in the SRS
- Identify each document by title, report number, date, and publishing organization.
- Specify the sources from which the references can be obtained.
- Include an entry for your project website.

Start of your text.

- [1] D. Thakore and S. Biswas, "Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks," Proceedings of IEEE Military Communication, Atlantic City, October 2005.



## 7 Point of Contact

For further information regarding this document and project, please contact **Prof. Daly** at University of Massachusetts Lowell (james\_daly at.uml.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.