Sample Format of Requirements Definition Document
Based on IEEE Standard 830: Software Requirements Specifications

1. Introduction Provide an overview of the complete document
   1.1. Purpose Define the purpose of the document and for whom it is written
   1.2. Scope
     ➢ Name of the product
     ➢ What the product will do
     ➢ What the product will not do
     ➢ Description of the benefits, goals, and objectives
   1.3. Definitions/Abbreviations
     ➢ Definition of all the terms
     ➢ Acronyms and abbreviations used in the document
       (Note: 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 requirement specification
   1.5. Overview Outline of the remainder of the document and its organization

2. General Description
   2.1. Product perspective
     ➢ Is this an independent product or is it part of a larger product?
     ➢ If it is part of larger product, identify the other components and describe the interfaces with those components
     ➢ Define the interfaces to other applications
     ➢ Identify the hardware to be used
   2.2. System Evolution
     ➢ Description of the life cycle model to be used
     ➢ Training setup intended
     ➢ Installation method intended
   2.3. Product Functions An overview of the functions of the system to be delivered
     (Note: Detailed discussion of the functions will be given in the Requirements Specification Document)
   2.4. Users of the product Identify the potential users of the product. For each category of user, describe general user characteristics relevant for the requirements specification. These include experience, training (educational level), technical expertise, and assigned priorities.
   2.5. General Constraints Indicate any constraints that apply. These may concern government regulations, hardware constraints, security regulations, and so on.
   2.6. 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.

3. Context diagram
4. Use case diagram
Sample Format of Requirements Specifications Document
Based on IEEE Standard 830: Software Requirements Specifications

This document contains all details which are relevant for the design phase to follow.

1. Analysis modeling
   1.1. Specify data entities and relationship between them using an Entity Relationship Diagram, or a Class Diagram.
   1.2. Specify the information flow and the transforms that are applied as data move from input to output using a Data Flow Diagram, or specify the dynamic behavior using Sequence Diagrams.

2. Functional requirements
   This subsection describes how the transformation of inputs to outputs is achieved. The following description is given for each individual function.
   2.1. Name of the function
   2.2. Description of the function A description of the purpose of this function.
   2.3. Inputs A precise description of the function’s inputs
      ➢ Source (how/where/who provides the input)
      ➢ Range of acceptable values
   2.4. Processing A definition of the operations that must be performed, such as checking for acceptable values, reaction to abnormal situations, or a description of the algorithms to be used.
   2.5. Outputs A precise description of the function’s outputs
      ➢ Destination (how/where/who receives the output)
      ➢ Error messages
   2.6. Pre-condition What must be true before the function is called
   2.7. Post-condition What is true after the function is called
   2.8. Requirements Indication of what other entities are used that are not directly input
   2.9. Side effects not directly related to output items
   2.10. Annotation
      2.10.1. Stability Will the needs change during the expected life of the product?
      2.10.2. Degree of necessity
         ➢ Mandatory – software will not be acceptable unless this function is provided in an agreed manner
         ➢ Desirable – this function would enhance the software product, but would not make it unacceptable if it is absent
         ➢ Optional – this function may or may not be worthwhile

3. Non-functional requirements
   3.1. External interface requirements
      3.1.1. User interfaces A description of the characteristics of the user interfaces, such as screen layout, function keys, help functions
      3.1.2. Hardware interfaces A description of the logical characteristics of hardware interfaces
      3.1.3. Software interfaces A description of software needed, such as a certain operating system or software package. Interfaces to other application software are also discussed here.
3.1.4. Communication interfaces An example is a communication protocol

3.2. 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 to be handled concurrently. Dynamic requirements concern operational performance of the system: how frequently will certain functions be called for and how fast should the system’s reaction be. These requirements should be stated in measurable terms,

3.3. Design constraints

3.3.1. Standard compliance Existing standards or regulations that must be followed, and the requirements that result from these.

3.3.2. Hardware limitations A description of the characteristics of the hardware, as far as they lead to software requirements.

3.4. Quality attributes In this section, particular attention is paid to quality aspects. These requirements must be measurable and verifiable.

3.4.1. Availability Factors that guarantee a certain level of availability, such as restart procedures, redundancy, etc.

3.4.2. Security Requirements regarding unauthorized access and others form of misuse. These include cryptographic techniques, constraints on the access and communication.

3.4.3. Portability Requirements regarding the portability of the software to different hardware platforms and/or different operating systems.

3.4.4. Maintainability Requirements to guarantee a certain level of maintainability of the system.

3.5. Other requirements A description of requirements that are specific to certain software, and which have not been discussed yet.
Sample Format of the Acceptance Test Plan

The purpose of the acceptance test plan is to describe the tests that must be performed to demonstrate all features of the product. For each function in your specification individual test plans should be developed for testing correctness and robustness. They should include data which exhibit both normal functionality and error conditions.