

Abstract: This document outlines projects requirements for the <project>. This is a controlled document and should be maintained in a configuration environment.

# Requirement Specification Document Template

[www.PMExamSmartNotes.com](http://www.PMExamSmartNotes.com)

---

## Contents

<b>REVISION HISTORY</b> .....	<b>1</b>
<b>INTRODUCTION</b> .....	<b>2</b>
PROJECT PURPOSE .....	2
PROJECT SCOPE .....	2
DEFINITIONS, ACRONYMS, AND ABBREVIATIONS USED .....	2
OVERVIEW .....	2
<b>HIGH LEVEL DESCRIPTION</b> .....	<b>2</b>
PRODUCT PERSPECTIVE .....	2
PRODUCT FUNCTIONS .....	2
USER CHARACTERISTICS .....	2
GENERAL CONSTRAINTS .....	3
ASSUMPTIONS AND DEPENDENCIES .....	3
<b>SPECIFIC REQUIREMENTS</b> .....	<b>3</b>
EXTERNAL INTERFACES .....	3
FUNCTIONAL REQUIREMENTS .....	3
<Functional Requirement or Feature #1> .....	3
<Functional Requirement or Feature #2> .....	3
NON-FUNCTIONAL REQUIREMENTS (NFR) .....	3
<i>Performance</i> .....	3
<i>Maintainability</i> .....	3
<i>Reliability</i> .....	4
<i>Availability</i> .....	4
<i>Security</i> .....	4
DESIGN CONSTRAINTS .....	4
LOGICAL DATABASE REQUIREMENTS .....	4
<b>DESIGN</b> .....	<b>4</b>
CLASS DIAGRAMS .....	4
SEQUENCE DIAGRAMS .....	4
STATE-TRANSITION DIAGRAMS .....	4
<b>CHANGE MANAGEMENT PROCESS</b> .....	<b>4</b>

## Revision History

Date	Description	Author	Version	Comments

## **Introduction**

<This should provide a complete overview of the software requirements specifications. The level of detailing should be enough for the designers to identify interfaces/modules and prepare high level design (HLD) of the modules.

This document should also explain the interaction with any third party interfaces and the nature, frequency and quality of interactions between these interfaces and the system.>

## **Project Purpose**

<This section should talk about the purpose of the project. >

## **Project Scope**

<You can use Project Scope template's content to fill in this section. This section outlines the overall scope of the project, and the products that project is going to produce.

Applications of this product, specific benefits, expected outcomes are outlined here.>

This section will also talk about exclusions – features that are not produced as part of this project.

## **Definitions, Acronyms, and Abbreviations Used**

<This section should talk about definitions specific to the project and domain, any acronyms and abbreviations to be used.>

## **Overview**

<Overview of this document, what does it cover, and how is it structured – are described here.>

## **High Level Description**

<Outlines high level description of this project and the product it produces – which are basically the product requirements.>

## **Product Perspective**

<This section puts the product into perspective with other related products. If the product is self-contained, it should be stated here. And if this is part of a larger system then interfaces between that system and this one needs to be outlined here.

A block diagram showing the major components of the larger system, interconnections, and external interfaces will be helpful.>

## **Product Functions**

<Outlines a brief description of all features as written from end-users' perspective. For example, for an Accounting system this section may talk about customer account maintenance, customer statement, and invoice preparation without going into minute details of these functions.>

## **User Characteristics**

<This section should outline the general characteristics of intended users of the product, such as education level, experience and technical expertise.>

## General Constraints

<Any constraints that will need to be kept in mind while designing and developing this product, are to be mentioned in this section. These include, but not limited to,

- Regulatory policies (specific to certain domains such as Healthcare)
- Hardware limitations
- Audit functions
- Control functions
- Reliability protocols >

## Assumptions and Dependencies

<This section should list each of the assumptions on requirements and any dependencies this product may have – including any on external products. Any impact on such assumptions and dependencies need to be mentioned as well.>

## Specific Requirements

<This section should detail the requirements to an extent that designers can pick it up from here and design subsystems.

Certain principles that apply while outlining requirements is that they need to be uniquely identifiable, organized to provide good readability, and related requirements need to be cross-referenced.>

## External Interfaces

<Dependencies on third party systems are to be mentioned here along with exact information being exchanged, protocols used, timing and frequency of communication, data formats, command formats etc.>

## Functional Requirements

<Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as “shall” statements starting with “The system shall...”>

<Functional Requirement or Feature #1>

<Functional Requirement or Feature #2>

## Non-Functional Requirements (NFR)

### Performance

<This subsection should specify both the static and the dynamic performance requirements placed on the software.>

### Maintainability

<This should specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc.>

### **Reliability**

<This should specify the factors required to establish the required reliability of the software system at time of delivery.>

### **Availability**

<This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.>

### **Security**

<This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure.>

### **Design Constraints**

<This section should specify design constraints that can be imposed by other standards, hardware limitations, etc.>

### **Logical Database Requirements**

<This section identifies entities and their relationships. The database needs, requirements and constraints are also mentioned.>

## **Design**

<Design approach is to be outlined here, including best practices, patterns and reusability.>

### **Class diagrams**

<This is a static design diagram that identifies high level components and derives low level components, their attributes and behaviour. >

### **Sequence Diagrams**

<This is a dynamic design diagram and shows flow of information from one point to another point.>

### **State-Transition Diagrams**

<This talks about how does state of a component changes as it passes through process flow in the system.>

## **Change Management Process**

<This section talks about the process to be used to make changes to this document.>