Version <XX>

Status < Draft/Approved >

Date: <dd/mm/yyyy>

Software Requirement Specification

<Project>

Disclaimers

The information contained in this document is the proprietary and exclusive property of <Organization name>. No part of this document, in whole or in part, may be reproduced, stored, transmitted, or used for design purposes without the prior written permission of <Organization name>.

The information contained in this document is subject to change without notice.

The information in this document is provided for informational purposes only. <Organization name> specifically disclaims all warranties, express or limited, including, but not limited, to the implied warranties of merchantability and fitness for a particular purpose, except as provided for in a separate software license agreement.

Document History

The following table shows the document history:

|  |  |  |
| --- | --- | --- |
| Date | Version | Comments |
|  |  |  |
|  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Intended Audience 4

1.3 Project Scope 4

1.4 References 4

2. Overview 5

2.1 Product Perspective 5

2.2 Product Features 5

2.3 User Classes and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 6

2.7 Assumptions and Dependencies 6

3. System Features 7

3.1 System Feature 1 (Name of the feature) 7

3.2 System Feature 2 (and so on) 7

4. External Interface Requirements 8

4.1 User Interfaces 8

4.2 Hardware Interfaces 8

4.3 Software Interfaces 8

4.4 Communications Interfaces 8

5. Non-functional Requirements 9

5.1 Performance Requirements 9

5.2 Safety Requirements 9

5.3 Security Requirements 9

5.4 Software Quality Attributes 9

6. Other Requirements 9

# Introduction

## Purpose

<The purpose of the product whose software requirements is covered in this document>

## Intended Audience

This document is intended for the following teams:

* Global Implementation
* Service Delivery
* Product Management
* Engineering
* Sales

## Project Scope

< A short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.>

## References

<List of documents or Web addresses to which this document refers including user guides, standards, system requirements specifications, use case documents.>

# Overview

## Product Perspective

<The context and origin of the product being specified in this SRS. For example, if this product is a member of any product family or a replacement of an existing system, or a new, self-contained product. If this SRS defines a component of a larger system, this section will mention the requirements of the larger system to the functionality of this software and identify interfaces between the two.>

<A simple diagram can be helpful here which displays the major components of the overall system, subsystem interconnections, and external interfaces.>

## Product Features

<A high-level summary of the major features of the product or the significant functions that it performs or allows the user to perform.>

<A top-level data flow diagram or a class diagram can be helpful in this section.>

## User Classes and Characteristics

<Information about the user classes who will use this product. The user class can be segregated depending on the frequency of use, a subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.>

## Operating Environment

<A description of the environment in which this software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Mention the limitations which might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<A list of assumptions and dependencies related to the product (for example, in the vision and scope document or the project plan).>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1 (Name of the feature)

3.1.1 Description and Priority

< A short description of the feature. The priority can also be assigned to the feature as High, Medium, or Low.>

3.1.2 Stimulus/Response Sequences

< A list of sequences of user actions and system responses that stimulates the behavior defined for this feature.>

3.1.3 Functional Requirements

<The functional requirements details associated with this feature. These are the software capabilities that must be present so that the user can carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

The functional requirements related to this feature are:

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# External Interface Requirements

## User Interfaces

<This section describes the logical characteristics of each interface between the software product and the users. The User Interfaces generally includes sample screen images, GUI standards, or product family style guides that are to be followed. This also includes screen layout constraints, standard buttons, functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed.>

## Hardware Interfaces

< This section describes the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

< This section describes the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. It also provides the data items or messages coming into the system and going out and describe the purpose of each. The Software interface section also provides data that will be shared across software components.>

## Communications Interfaces

< This section describes the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. This also provides the information of any communication standards that will be used, such as FTP or HTTP>

# Non-functional Requirements

## Performance Requirements

<This section specifies the performance requirements for the product under various circumstances and explain their rationale, to help the developers understand the intent and make suitable design choices. The performance requirements section also provides the timing relationships for real-time systems. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<This section specifies the requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. This section also provides information about any safeguards or actions that must be taken, as well as actions that must be prevented.>

## Security Requirements

< This section specifies the requirements regarding security or privacy issues surrounding the use of the product or protection of the data used or created by the product. This section also provides information regarding any user identity authentication requirements.>

## Software Quality Attributes

< This section specifies any additional quality characteristics for the product that will be important to either the customers or the developers. These attributes are - adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability.>

# Other Requirements

<This section provides any other requirements apart from the requirements mentioned in the above sections of the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on.>

Appendix A: Glossary

< All the terms necessary to properly interpret the SRS, including acronyms and abbreviations.>

Appendix B: Analysis Models

<Includes any relevant analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< A list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and so on.>