**SOFTWARE REQUIREMENTS SPECIFICATION**

***for***

**[PROJECT NAME]**

***Prepared By***

[Grup Name]

[Grup Members]

[Due Date]

CHANGE HISTORY

*[The change history shall include the project name, version number of the report, date of release and a list of version numbers and dates of release of all previous versions of the report.]*

PREFACE

*[The preface of the Software Requirements Specification shall describe the scope and context of the Software Requirements Specification.]*

**TABLE OF CONTENTS**

[CHANGE HISTORY i](#_Toc528078218)

[PREFACE ii](#_Toc528078219)

[TABLE OF CONTENTS iii](#_Toc528078220)

[LIST OF FIGURES v](#_Toc528078221)

[LIST OF TABLES vi](#_Toc528078222)

[1 INTRODUCTION (Section 1 of the SRS) 1](#_Toc528078223)

[1.1 Purpose of the SRS 1](#_Toc528078224)

[1.2 Scope of the Product 1](#_Toc528078225)

[1.3 Definitions, acronyms and abbreviations 1](#_Toc528078226)

[1.4 References 1](#_Toc528078227)

[1.5 Overview 1](#_Toc528078228)

[2 OVERALL DESCRIPTION (Section 2 of the SRS) 2](#_Toc528078229)

[2.1 Product perspective 2](#_Toc528078230)

[2.1.1 System interfaces 2](#_Toc528078231)

[2.1.2 User interfaces 2](#_Toc528078232)

[2.1.3 Hardware interfaces 2](#_Toc528078233)

[2.1.4 Software interfaces 2](#_Toc528078234)

[2.1.5 Communication interfaces 2](#_Toc528078235)

[2.1.6 Memory Constraints 2](#_Toc528078236)

[2.1.7 Operations 3](#_Toc528078237)

[2.1.8 Site adaptation requirements 3](#_Toc528078238)

[2.2 Product Functions: 3](#_Toc528078239)

[2.3 User Characteristics: 3](#_Toc528078240)

[2.4 Constraints: 3](#_Toc528078241)

[2.5 Assumptions and Dependencies: 4](#_Toc528078242)

[2.6 Apportioning of Requirements: 4](#_Toc528078243)

[3 SPECIFIC REQUIREMENTS (Section 3 of the SRS) 5](#_Toc528078244)

[3.1 External Interfaces: 5](#_Toc528078245)

[3.2 Functions: 6](#_Toc528078246)

[3.3 Performance Requirements: 6](#_Toc528078247)

[3.4 Logical Database Requirements 7](#_Toc528078248)

[3.5 Design Constraints 7](#_Toc528078249)

[3.5.1 Standards Compliance 7](#_Toc528078250)

[3.6 Software System Attributes 7](#_Toc528078251)

[3.6.1 Reliability 7](#_Toc528078252)

[3.6.2 Availability 7](#_Toc528078253)

[3.6.3 Security 7](#_Toc528078254)

[3.6.4 Maintainability 7](#_Toc528078255)

[3.6.5 Portability 7](#_Toc528078256)

LIST OF FIGURES

*[List the name and page number of the figures used in the report]*

LIST OF TABLES

*[List the name and page number of the tables used in the report]*

# INTRODUCTION (Section 1 of the SRS)

## Purpose of the SRS

In this subsection;

1. The purpose of the SRS
2. The intended audience of the SRS

will be specified.

## Scope of the Product

In this subsection, give the name of your software product, then

1. Explain what the software product will, and, if necessary will not do
2. Describe the application of the software being specified, including relevant benefits, objectives, and goals.

## Definitions, acronyms and abbreviations

In this subsection, all the necessary definitions and abbreviations required to properly interpret the SRS will be specified.

## References

In this subsection, a complete list of all documents referenced in the SRS will be specified.

## Overview

In this subsection, the organization and the content of the rest of the SRS will be specified.

# OVERALL DESCRIPTION (Section 2 of the SRS)

Describe the general factors that affect the product and its requirements. Don’t state specific requirements, instead provide a background for those requirements.

## Product perspective

### System interfaces

If the software is independent and totally self-contained, it should be stated here. Otherwise (if the proposed software is part of a larger system), list each system interfaces and identify the functionality of the software to accomplish the requirements.

### User interfaces

In this subsection, the characteristics of each interface between the software product and its users (e.g. required screen formats, page or window layouts, and content of any report or menus) and the aspects of optimizing the interface with the user will be specified.

### Hardware interfaces

In this subsection, the characteristics of each interface between the software product and the hardware components of the system will be specified. What devices are to be supported and how they are supported will be identified.

### Software interfaces

In this subsection, the use of other required software products and interfaces with other application systems will be specified. For each required software product, the following should be provided in a single table or separately:

1. Name
2. Version number
3. Source
4. Purpose of the interfacing software with the proposed software.

### Communication interfaces

In this subsection, various interfaces to communications will be specified such as local network protocols.

### Memory Constraints

This should specify any applicable characteristics and limits on primary and secondary memory.

### Operations

This should specify the normal and special operations required by the user such as:

1. The various modes of operations in the user organization for example user-initiated operations. (In order to show the user required operations you can use **Use-Case diagrams**)
2. Periods of interactive operations and periods of unattended operations
3. Data processing support functions
4. Backup and recovery operations.

### Site adaptation requirements

This should;

1. Define the requirements for any data or initialization sequences that are specific to a given site, mission or operational mode such as grid values and safety limits;
2. Specify the site or mission-related features, which should be modified to adapt the software to particular installation.

## Product Functions:

This subsection of SRS should provide a summary of the major functions that the software will perform. Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product. Note that for the sake of clarity

1. The functions should be organized in a way that makes the list of functions understandable to the acquirer.
2. Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product. However, simply shows the logical relationships among variables.

## User Characteristics:

This subsection of SRS should describe those general characteristics of the intended users of the product including educational level, experience and technical expertise. It should not be used to state specific requirements, but rather provide the reasons why certain requirements are later specified in section 3 of the SRS.

## Constraints:

This subsection of SRS should provide a general description of any other items that will limit the developer’s options. These include:

1. Regulatory policies,
2. Hardware limitations
3. Interface to other applications
4. Parallel operation,
5. Audit functions,
6. Control functions,
7. Higher-order language requirements,
8. Signal handshake protocols,
9. Reliability requirements,
10. Criticality of the application,
11. Safety and security considerations.

## Assumptions and Dependencies:

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not designed constraints on the software. However, they are any changes to them that can affect the requirements in the SRS.

## Apportioning of Requirements:

This subsection of SRS should identify requirements that may be delayed until versions of the system.

# SPECIFIC REQUIREMENTS (Section 3 of the SRS)

This subsection of SRS should contain the software requirements to a level of detail sufficient to designer to design a system to satisfy those requirements and testers to test that system satisfies those requirements. Every stated requirement should be externally perceivable by users, operators or other external systems. **These requirements should include at a minimum a description of every input into the system, every output from the system, and all functions performed by the system in response to an input or in support of an output.** As this is often the largest and most important part of the SRS, the following principles apply:

1. Specific requirements should be sated in conformance with all the following characteristics
   * Correct,
   * Unambiguous,
   * Complete,
   * Consistent,
   * Ranked for importance and stability,
   * Verifiable,
   * Modifiable,
   * Traceable.
2. Specific requirements should be cross-referenced to earlier documents that relate.
3. All requirements should be uniquely identifiable.
4. Careful attention should be given to organizing the requirements to maximize readability.

## External Interfaces:

This should be detailed description of all inputs into and outputs from the software system. It should complement the interface description in section 2. It should include both content and format as follows:

1. Name of item,
2. Description of purpose,
3. Source of input or destination of output,
4. Valid range, accuracy, and/or tolerance,
5. Units of measure,
6. Timing,
7. Relationships to other inputs/outputs,
8. Screen formats/organization (You can give a draft screen view),
9. Window formats/organization,
10. Data Formats,
11. Command formats,
12. End messages.

## Functions:

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

1. Validity checks on the inputs,
2. Exact sequence of operations,
3. Responses to abnormal situations, including
   * Overflow,
   * Communication facilities,
   * Error handling and recovery,
4. Effect of parameters,
5. Relationship of outputs to inputs, including,
   * Input/output sequences,
   * Formulas for input to output conservation.

It may be appropriate to partition the functional requirements into sub functions or sub processes. This does not imply that the software design will also be partitioned that way. (You can use Data Flow Diagrams or Use Cases.)

## Performance Requirements:

This subsection should specify both the static and the dynamic numerical requirements placed on the software or human interaction with the software as a whole. Static numerical requirements may include the following:

1. The number terminals to be supported,
2. The number of simultaneous users to be supported,
3. Amount and type of information to be handled.

Static numerical requirements are sometimes identified under a separate section entitled Capacity.

Dynamic numerical requirements may include, for example, the number of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

All of these requirements should be in measurable terms. For example: 95% of the transactions shall be processed in less than 1 s. rather than an operator shall not have to wait for the transaction to complete.

**Note:** Numerical limits applied to one specific function are normally specified as a part of processing subparagraph description of that function.

## Logical Database Requirements

This should specify the logical requirements for any information that is to be placed into a database (You can use ER diagram).

## Design Constraints

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

### Standards Compliance

This subsection should specify the requirements derived from existing standards or regulations. They may include the following:

* Report format
* Data naming
* Accounting procedures
* Audit tracing

## Software System Attributes

### 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.

### Security

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

### Maintainability

This should specify attributes of software that relate to the ease of maintenance of the software itself.

### Portability

This should specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems.