Template No.
CEMILAC_SYSGP_SRD_06

Software Requirement Standard of <LRU/System Name> for <Platform Name>

Issue/Rev No: 01/00
Date of Release: 8 Feb 2025

<DESIGN AGENCY LOGO>	**Document No.:**			
	Issue No. :	<00X>	**Issue Date :**	<DD/MM/YYYY>
	Copy No. :	01 of N	**No. of Pages :**	<Total No. of Pages >
	Document Classification :	□Secret □Restricted	□Confidential □Unrestricted	

Title:	**Project/System :**
Software Requirements Standard **of** **<LRU/SYSTEM Name>for <Platform name>**	**<System/Project Name>**
	LRU/System Part No.:
	<No.>
	Software Criticality Level :
	DO-178C Level <A/ B/ C/ D>

	Name & Designation	**Signature**
Prepared By	<Design Rep Name>, < Designation>	
Reviewed By	<Project Leader Name>, <Designation> <AWG/QA HOD Name>, <Designation>	
Approved By	<RCMA Officer>, <Designation> <RCMA Name>	

<Design Agency Name & Address>

Disclaimer:

This document is a guidance document. Applicable section / table rows may be considered. Any additional details may be added. Any not applicable section/ table rows may be deleted. The template is very general and vary with process to process followed by Development Agency. The document may be fine-tuned with the TAA for finalization.

Distribution List

Copy No.	Designation of the Copy Holder	Organisation
01	Head of Design Agency	Design Agency Name
02		
03		
04		

Amendment History

Issue, Rev. No.	Issue, Rev Date	Change Request Ref.	Brief Description of Amendment	Affected Pages/ Section	Changed By	Change Effective Date
01, 00		NA	Initial Issue	NA	NA	Initial

Table of Contents

1. Introduction

Software Requirements Standards defines the methods, rules, and tools to be used to develop the high-level requirements.

1.1 Purpose

This document is used to demonstrate that high-level requirements are developed for <LRU_NAME>, and that the derived high-level requirements are indicated to the system safety assessment process. This is one of the major lifecycle data to be generated during the planning phase of the project in accordance with Section 4.5 and 11.9 of RTCA/DO-178C.

1.2 Scope

This document describes, for <LRU_NAME>, the description of system requirements allocated to software, modes of operations and functional and non-functional requirements under each, performance criteria, timing and memory constraints, interface details, and failure modes. The software is being developed at <Deign Agency_Name> for the End User <Customer Name>, Ministry of Defence.

1.3 Applicable Documents

Define the list of all applicable documents in following sections:

1.3.1 External documents

Define the list of all applicable documents of external origin, relevant for this project.

1.3.2 Internal documents

Define the list of all applicable documents of internal origin, relevant for this project.

1.4 Part Number and Nomenclature

Define the details of all software components having unique part number and nomenclature to identify them through the software development life cycle.

1.5 Acronyms and Abbreviations

Define all the abbreviations and acronyms are listed with their expanded names.

2. System Overview

This section provides an overview of the system, including description of its functions along with block diagram.

3. Software Overview

This section provides an overview of the software, including description of its all major functions.

4. Methods for Software Requirements

4.1 System Requirements

System requirements are identified from customer-provided system-level documents. System requirements must be mapped to either hardware requirements or software requirements. If mapped to software requirement, it may be further mapped to a specific CSCI if there are more than one CSCI in the sub-system.

4.2 High Level Requirements

Once the system requirements are mapped to CSCI specific software requirements, define all High- level requirements that have direct traceability to the system requirement.

4.3 Derived High Level Requirements

Define the derived high level requirements that have direct or indirect traceability to the high level requirements or system requirements.

Provide all derived requirements as input to the System Safety Assessment process.

4.4 Non- Functional Requirements

4.4.1 Interface Requirements

Define external interface requirements, which interacts with CSCI. Provide details of the interface, such as communication protocol, signal characteristics, periodicity etc.

4.4.2 Safety Requirements

Define the requirements that are responsible for the security aspect of the software system including safety strategies, design constraints and design methods, such as, partitioning, dissimilarity, redundancy, or safety monitoring.

4.4.3 Performance Requirements

Define specific performance requirements of the Software.

4.4.4 Unique Identification

Define unique identification to each requirement.

4.4.5 Security Requirements

Define specific security requirements of the Software.

4.4.6 Maintenance Requirements

Define specific maintenance requirements of the Software.

4.4.7 Traceability

Provide traceability for the software requirements from System Requirements to SRD and backward traceability.

4.4.8 Assumptions and Constraints

List out the assumptions and constraints. If certain details cannot be specified, mention as TBD. At an early opportunity, validate the assumptions, and convert them to facts.

4.4.9 Additional Requirements

Define additional requirements (if any) of the software. If any requirement is not clear, define it as' TBD'.

5. Methods for Derived Requirements

Define the methodologies or techniques that are used for identifying, documenting and integrating derived requirements.

6. Notations:

Define all the Symbols, diagrams and textual convections for high level requirements by using UML diagrams (as applicable) as below:

a) Scenarios to describe typical usage of the system; scenarios shall be written in natural language, with application-domain terminology.

b) Use Case Diagrams to represent the functionality.

c) Class Diagrams to represent the structure.

d) Sequence Diagram to formalize the dynamic behaviour and to visualize communication between objects.

e) State Chart Diagrams to describe dynamic behaviour of the system and/or individual objects as number of states and transition between these states.

f) Provide Unique Id to each requirement so that it may be traced throughout the software life cycle.

Standard UML Notations or others may be used for all of these diagrams.

Each use case diagram shall have an associated 'ticket' describing its priority, preconditions input, outputs, basic and exceptional behavior, and any design constraints.

7. Tools for Requirements

Tools are used for requirement capturing process, requirements development, manage and document software requirements. It is used to maintain the consistency, compatibility and efficiency thought the design process.

<Define tools to be used for developing SRD document>.

7.1Tools to be used

<Define list of tool(s) to be used for development of SRD document, UML diagrams etc>

7.2 Constraints on Requirements Development Tools

<Define any limitation(s) or any constraint(s) on the use of the tools used.>