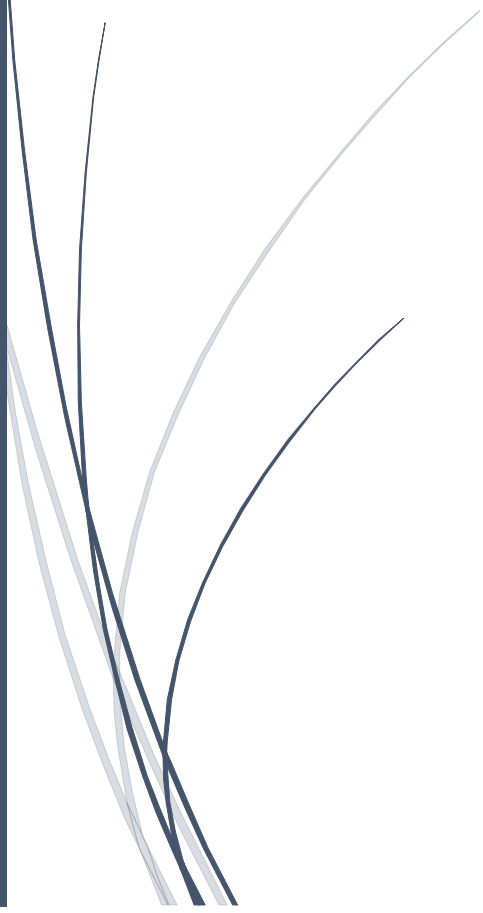
Template No.
CEMILAC_SYSGP_SDD_10

Software Design Description

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	☐ Confidential	
		☐ Restricted	☐ Unrestricted	

Title:	**Project/System :**
Software Design Description **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	Organization
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 Design Standard (SDD) is description of software architecture and the low-level requirements that will satisfy the high-level requirements.

1.1. Purpose and Scope:

This document provides the detailed description of the implementation of the architecture and low-level requirement, also with compliance to the High-level requirements. This document determines the details for the Design Phase.

1.2. Applicable Documents

Define the list of all applicable documents in following sections:

1.2.1. External Documents

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

1.2.2. Internal Documents

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

1.3. 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.

1.4. Acronyms and Abbreviations

Define all the abbreviations and acronyms with their expanded names in this section.

e.g.	CSCI	Computer Software Configuration Item
	CSC	Computer Software Component

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. Preliminary Design

4.1 CSCI Overview

<Define major functionalities of CSCI and communication of CSCI with various external interfaces during its execution>.

4.1.1. CSCI Architecture

<Define the Software Architecture i.e. Division of functionality into lower level components and inter-relationship between them. Provide architectural / decomposition diagram.

<Define the constituent layers in the hierarchy are described below>:

Application Layer: The software components present in this layer are a direct representation of the customer problem domain. It being the highest level is independent of the hardware. Thus, any change made on the hardware does not require a change in the software present in this layer.

System Services Layer: The software components present in this layer map the services the application requires to the services provided by the hardware. This layer is a bridge between the uppermost and lowermost layers in the architecture.

Resource access Layer: The components of this layer are abstractions of the real hardware. The software in this layer is largely affected in the event of a hardware change.

4.1.2. System Modes

<Define all the various possible modes of LRU e.g. Record Mode, Maintenance Mode etc.>

4.1.3. Control and Data flow

Define Diagrammatic representation of control and data flow from booting of software on power-ON to the acquisition of inputs, processing the information, producing the outputs until power-off/ shut down/ end of operation.

<Define State diagrams/ activity diagrams/ transition charts may be used for this purpose>.

4.1.4. Memory and Timing Requirements

<Define all memory requirements in KB, MB, GB e.g. FLASHRAM for storage of application program in, RAM for running the program, Data Storage etc>

<Define timing requirements for specific requirement which are to be followed strictly during implementation>

4.1.5. Scheduling

<Define scheduling procedures and inter-processor/inter-task communication mechanisms, including time-rigid sequencing, pre-emptive scheduling and interrupts. How the timing requirements are met>

4.2 Design Description of <CSCI Name>

<Define UseCase or CSC Name, Identifier No, Purpose, details of classes/functions to be included in design or external classes/functions which shall be linked with it, Design Constraints, traceability link w.r.t. to SRD document etc>

Sample Example:

CSC Name **:** Acquisition of Aircraft Parameters

Identifier **:** <CSCI_Name>_UC_03

Purpose : Acquires Analog, Discrete parameters @8 Samples/Secs.

Classes/Functions **:** This CSC consists of the following classes:

Identified Class	*Layer*	*Scope*
CDataAcquisition	*System Service Layer*	*Manages the acquisition operation*

Design Constraints **:** <Define Design Constraints (if any)>

Traceability **:** This CSC satisfies the requirements as per sec N.n of SRD.

<If there are more than one CSCI e.g. CSCI for Main Processor Board, CSCI for Communication Board etc, separate or common Design document may be prepared based on volume of source code>

5. Detailed Design of <CSCI Name>

<Define the details of class/function in details, so that source code may be developed inline of design descriptions>.

5.1 <Class/Function Name>

<Write brief description of Class/Function>

<Define **List of variables**, global variables i.e. Variable Name as per Coding Standard Rules, its type, Description/usage in tabular form>

<Define **List of functions** i.e. function Name as per Coding Standard Rules, Return Type, Description/usage in tabular>

<Define **brief description of function**>

< Define **function prototype**>

< Define **Input and output** of each function>

< Define **step-by-step design details**/ pseudo code/ logic flow in structured English or graphically in a flow chart to describe internal design details of each of the components of architectural diagram including conditions for decision making, formulas used, algorithms used, built-in robustness etc >

6. Derived Requirements

<Provide the details of derived requirements which were generated due to specific hardware/ user due to design process. These may not affect the externally visible functionality of the software and also, these being design dependent, and should not affect the intended functions of system, if the design is changed.

<Details may be provided in tabular form with relevant section of preliminary design or of detail design>.

7. Safety and Security Concerns

< Define description of the means to ensure that the code cannot be enabled in the target computer, if the system contains deactivated code>

<Rationale for those design decisions that are traceable to safety-related system requirements>. <Partitioning methods and means of preventing partition breaches>. Also define design details for fault tolerance, functional partitioning requirements (as applicable).

8. List of Annexure

Annexure-A Activity Diagrams

<Define all activity Diagrams>

Annexure-B Sequence Diagrams

<Define all Sequence Diagrams>

Annexure-C Class Diagrams

<Define all Class Diagrams>

<Include all other relevant diagrams in annexure and give reference in the main body of document.>