**TABLE OF CONTENTS**

[INTRODUCTION](#_knalhjf9o59u)

[Purpose](#_afv4qstxd1lu)

[Scope](#_77k1pfuhc78j)

[Definitions, Acronyms, and Abbreviations](#_47qw9ki7a4aq)

[References](#_kfqrjeyo0eoz)

[SYSTEM ARCHITECTURAL DESIGN](#_id2v88s75wb9)

[System Description](#_eeh75nq0lj90)

[System Architecture](#_9eysw0va7301)

[Design Constraints (optional)](#_gg5082mluktj)

[General constraints](#_t4fvgp4ngdm)

[Hardware constraints](#_3e0k8epvtkhc)

[SW Constraints](#_a4gpi2ffwr5b)

[Components description](#_475g2urhxj10)

[Introduction](#_rrfyjpajsqv4)

[Decomposition description](#_x5lce5crwk2e)

[Component 1..](#_qjb909gl9eob)

[External interfaces](#_rxs0mkx9bldr)

[Introduction](#_k0ok7rl9dyzq)

[User interfaces](#_rtcuwdnwydg0)

[Interface 1..](#_lf1hn61jcqit)

[External system interfaces](#_4lkq9dmr1247)

[Interface 1..](#_98ymsa2i4cqm)

[Detailed design](#_4x2zzxjdwq1a)

[Introduction](#_aniggb4gllup)

[Component 1…](#_qlx4dt974ycl)

[Annexes](#_3w3pkvz3nq8z)

[Traceability](#_gk7z2ox5oiq3)

[IEEE 1016 and UML mapping](#_kdmfwdrc4gvc)

# INTRODUCTION

## Purpose

*Provide an overview of the entire document.*

## Scope

*Provide the document scope and intended audience*

## Definitions, Acronyms, and Abbreviations

*All non-standard terms, acronyms and abbreviations that are unique to this document should be included in this section. References to other appendices or reference documents may be included. Index information should be in bold.*

* ***IEEE*** *– Institute of Electrical and Electronics Engineers*
* ***SRS*** *– System Requirements Specification*
* ***PRD*** *– Product Requirements Document*
* ***MRD*** *– Marketing Requirements Document*

*Also see IEEE Std 1002-1987, IEEE Standard Taxonomy for Software Engineering Standards.*

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
|  |  |
|  |  |

# References

*Provide references for any pertinent document.*

# SYSTEM ARCHITECTURAL DESIGN

*Provide a general description of the system to be designed as well as the design and development methodology.*

## System Description

*Give a general description of the complete system here; preferably in form of a commented drawing. This is only for an overview and can be left out or be the same as used in the SW specification document.*

## System Architecture

*Describe the chosen system architecture. For instance, architectural design patterns can be used to describe the system: peer-to-peer, client/server, stand-alone or embebbed systems, etc.*

*Include also discussion or description of alternative designs*

## Design Constraints (optional)

### General constraints

*Describe the general constraints implied by the design process and what are the impacts on the system architecture and modules design (time, tools, resources, etc.).*

### Hardware constraints

*Describe constraints imposed by limited hardware resources.*

### SW Constraints

*Describe constraints imposed by software environment (e.g. OS, languages, etc.)*

# Components description

## Introduction

*Describe the system decomposition strategy guided by the use case view of the system (requirement and analysis included in previous-related documents). In the use case view, the system is studied as a black box. In this document this black box will be decomposed following a specific philosophies or design patterns (e.g. Model-View-Controller, J2EE design patterns, etc.). This design strategy has to be described here.*

## Decomposition description

*Describe how the system is divided into partitions represented by design entities and describe the important properties and relationships among those entities.*

*The objective is to divide the system into separate components that can be considered, implemented, changed, and tested with minimal effect on other entities. Entities can exist as a system, subsystems, data repositories, packages, modules (classes), and processes (concurrency).*

*Blueprints of the system have to be used to describe the structure. For instance, UML structure diagrams can be used (e.g. class/object, composite, components, package, deployment diagrams).*

*In the next section, the description of every component has to include dependency (relation with others) and interface views.*

*Note: if required, hierarchies or categories could be used to group components (e.g. packages, data, process, etc..)*

### Component 1..

*Each component or design entity will have common information to be specified: name, purpose, and function. There are common relationships among entities such as interfaces or shared data. The common characteristics of entities are described by design entity attributes.*

# External interfaces

## Introduction

## User interfaces

*Describe interfaces with the various system users. Even if this information could already be described in previous related documents (i.e. SRS or analysis documents), in this section the description of this interfaces could be more specifically done as the blueprints to be followed by designers, programmers and testers.*

### Interface 1..

## External system interfaces

*Describe the interfaces with external system or software interacting with the system.*

*In this section, designers, programmers, and testers should find important information about design entities that they did not develop. These entities may be reused from earlier projects, contracted from an external source, or produced by other developers. The interface description provides the contract or agreement among designers, programmers, and testers about how cooperating entities will interact. A clear description of entity interfaces is essential on a collaborative development process to facilitate the integration and maintenance.*

### Interface 1..

# Detailed design

## Introduction

*This section provides the internal details of each design entity. For every component, a general description needs to be provided followed by a detailed behavior specification. This specification can be written using visual or textual notations (e.g. UML or textual algorithms) and needs to include passive (methods, functions or procedures) and/or active behaviour (state machine or activity diagrams).*

### Component 1…

# Annexes

*This chapter includes additional documentation. For instance:*

## Traceability

*In this section a treaceaililby table must be included. This section is very helpful for navigating through the documentation, linking requirement specification (with its functional view) to the system design (where the implementation of all functions and the means to do so are described in the “structural view“). It can also link the test plan for functional testing to the SW specification.*

## IEEE 1016 and UML mapping

This section should be deleted from the final document, it is only included here for helping designers to use UML specification within this document standard. Next figure illustrates how different UML diagrams could be used to illustrate the different views of the system design.

