System XXX

Detail design of xxx

**Document Code:**

**XXX-XXX**

**Changes History**

Change Type: A - Added M - Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Implementer** | **Change Type** | **Detail** | **Version** |
| dd/MM/yyyy | Abc.ho | A | Khởi tạo tài liệu | 1.0 |
|  |  |  |  |  |

TABLE OF CONTENTS

1 Introduction 1

1.1Document overview 1

1.2References 1

1.2.1 Project References 1

1.2.2 Standard and regulatory References 1

2 Software Architecture overview 2

3 Software design description 2

3.1Component 1 2

3.1.1 Component interfaces 2

3.1.2 Component design description 2

3.1.3 Workflows and algorithms 2

3.1.4 Software requirements mapping 2

3.2Component 2 2

3.2.1 Component interfaces 2

3.2.2 Component design description 2

3.2.3 Workflows and algorithms 3

3.2.4 Software requirements mapping 3

3.3Component 3 3

3.3.1 Component interfaces 3

3.3.2 Component design description 3

3.3.3 Workflows and algorithms 3

3.3.4 Software requirements mapping 3

4 Critical Requirements 3

# Introduction

You may have all the description of the design of your software in a single instance of this document or have the design of each component/package/element in many instances of this document. This is your choice, which depends on the size of your software.

## Document overview

This document describes the design of XXX component/package/element of XXX device, part of XXX software development project.

## References

### Project References

| # | Document Identifier | Document Title |
| --- | --- | --- |
| [R1] | ID | Add your documents references.  One line per document |

### Standard and regulatory References

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| [STD1] |  | Add your documents references.  One line per document |

Add the standard references to the table above. It may include ISO 14971, ISO 13485, IEC/TR 80002-1, IEC 62304, amongst others.

# Software Architecture overview

Describe here the top level software components and their interactions/relationships.

Use UML package diagrams and/or layer diagrams and/or interface diagrams.

Describe also the operating systems on which the software runs.

You may reference the system architecture document, if you have one in your project, which already explains the software architecture.

# Software design description

If you have a single design document, describe each top level package/component of your software and if necessary sub-components/sub packages.

If you have one design document for each top level package/component, describe the content (sub components, sub packages) of each top level package/component.

Use Class diagrams, Collaboration / sequence diagrams, Deployment diagrams to illustrate your description.

## Component 1

### Component interfaces

Describe the interfaces of the component and input output data

### Component design description

Describe the design of the component, Use package diagrams and/or class diagrams to show the links between sub-components/sub-packages and or classes inside the component.

### Workflows and algorithms

Use collaboration diagrams  and/or sequence diagrams to show the workflows of components/packages/classes inside the component.

Describe algorithms, if possible. An algorithm may be described outside this document, in this case, add the reference to that document.

### Software requirements mapping

Class C software only, list the SRS requirements handled by this component

## Component 2

Repeat the pattern for each component.

### Component interfaces

Describe the interfaces of the component and input output data

### Component design description

Describe the design of the component, Use package diagrams and/or class diagrams to show the links between sub-components/sub-packages and or classes inside the component.

### Workflows and algorithms

Use collaboration diagrams  and/or sequence diagrams to show the workflows of components/packages/classes inside the component.

Describe algorithms, if possible. An algorithm may be described outside this document, in this case, add the reference to that document.

### Software requirements mapping

Class C software only, list the SRS requirements handled by this component

## Component 3

Repeat the pattern for each component.

### Component interfaces

Describe the interfaces of the component and input output data

### Component design description

Describe the design of the component, Use package diagrams and/or class diagrams to show the links between sub-components/sub-packages and or classes inside the component.

### Workflows and algorithms

Use collaboration diagrams  and/or sequence diagrams to show the workflows of components/packages/classes inside the component.

Describe algorithms, if possible. An algorithm may be described outside this document, in this case, add the reference to that document.

### Software requirements mapping

Class C software only, list the SRS requirements handled by this component

# Critical Requirements

If requirement were tagged as critical in the SRS, add here the traceability between these requirements and the components described in this document. Critical requirement may be those added after risk analysis.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement title | Component | Comment |
| REQ-001 | Software shall have an abort button | Main window | Widget added in the window layout |
|  |  | Main window controller | Controller aborts the operation |