Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

1 Introduction 3

1.1 References 3

1.1.1 Project References 3

2 Software Architecture overview 3

3 Software design description 3

3.1 Component 1 3

3.1.1 Component interfaces 3

3.1.2 Component design description 3

3.1.3 Workflows and algorithms 3

3.1.4 Software requirements mapping 3

3.2 Component 2 3

3.2.1 Component interfaces 3

3.2.2 Component design description 4

3.2.3 Workflows and algorithms 4

3.2.4 Software requirements mapping 4

3.3 Component 3 4

3.3.1 Component interfaces 4

3.3.2 Component design description 4

3.3.3 Workflows and algorithms 4

3.3.4 Software requirements mapping 4

4 COTS Identification 4

# Introduction [EREN]

Brief description of the software system and the purpose of the document.

This document describes the design of the XXX software system.

## References

### Project References

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

# Software Architecture overview [EREN]

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

Use UML diagrams.

# Software design description

Describe each top level package/component of your software and if necessary sub-components/sub packages.

Use Class diagrams, sequence diagrams and deployment diagrams to illustrate your description.

## Graphical User Interface [ERDI]

### Component interfaces

Describe the interfaces of the component and input output data

### Component design description

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

### Workflows and algorithms

Use sequence diagrams and activity 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

List the SRS requirements handled by this component

## Controller [EMRE, ERDI]

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 class diagrams to show the links between sub-components/sub-packages and or classes inside the component.

### Workflows and algorithms

Use 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

List the SRS requirements handled by this component

## Network [DENIZ, EREN]

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 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

List the SRS requirements handled by this component

# COTS Identification [ALL]

List external software components/libraries that your system rely on, if there are any.

Example:

COTS (commercial of the shelf) libraries used in XXX are the following:

* foo.jar, version id, download URL, License type,
* bar.jar, version id, download URL, License type,