|  |  |
| --- | --- |
| Artifact Name | **Software Design Document** |
| Artifact Code | CMS\_SoftwareDesign |
| Artifact Version | Draff |
| Project Name | VNN-CMS |
| Project Code | VNN-CMS |

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 28-05-2008 | Draff | Initiate | Duong Hai Phong |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

TABLE OF CONTENT

[1. Introduction 3](#_Toc176080834)

[1.1. Purpose 3](#_Toc176080835)

[1.2. Scope 3](#_Toc176080836)

[1.3. Overview 3](#_Toc176080837)

[2. Common Package and Mechanism 3](#_Toc176080838)

[2.1. Common package 3](#_Toc176080839)

[2.2. Error, exception handling 3](#_Toc176080840)

[2.3. Log, trace and debug 3](#_Toc176080841)

[2.4. Performance optimizing mechanism 3](#_Toc176080842)

[2.5. Multilingual processing 3](#_Toc176080843)

[3. Diagrams 3](#_Toc176080844)

[3.1. Customer management 3](#_Toc176080845)

[4. Packages 3](#_Toc176080846)

[4.1. xxx Package 3](#_Toc176080847)

[5. Appendix 3](#_Toc176080848)

[5.1. Abbreviations and acronyms 3](#_Toc176080849)

[5.2. References 3](#_Toc176080850)

[6. Annexes (Optional) 3](#_Toc176080851)

## Introduction

### Purpose

### Scope

### Overview

## Common Package and Mechanism

### Common package

#### Class diagram

[Class diagram]

|  |  |  |
| --- | --- | --- |
| No | Class Name | Description |
| 01 | <Name of class> | <Brief description about class ex. One sentence to tell what the class is for, what does it encapsulate> |

#### XXX class

[Class description]

* Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | <Attribute name> | <Int> |  | <Public/ Static> | <Description of attribute> |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | <method name> | <brief description of method. can be one sentence tell what the method does> |
|  |  |  |
|  |  |  |

* xxxx method

[Method declaration]

[method description, it must be compliance with the brief description in the upper class list]

* Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | <parameter name> |  |  |  | <Description of parameter, special criteria such as boundary should be stated> |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* Implementation

[How to implement the method, it can be in pseudo code or activity diagram or just words]

### Error, exception handling

#### Class Diagram

[Describe class like in common package]

#### Usage mechanism

[Common mechanism of exception handling]

### Log, trace and debug

### Performance optimizing mechanism

### Multilingual processing

## Diagrams

[Describe diagrams in system such as collaboration diagram, sequence diagram, activities diagram and state chart for some functionalities of the system]

[Example]

### Customer management

#### Add customer



**Figure 1 Add Customer sequence diagram**



#### Update customer

.....

## Packages

|  |  |  |  |
| --- | --- | --- | --- |
| No | Package | Language | Description |
| 01 | <package name> | C++, Java | <brief description of package; can be one sentence tell what the method does> |
|  |  |  |  |
|  |  |  |  |

### xxx Package

#### Class diagram

[Class diagram figure]

|  |  |  |
| --- | --- | --- |
| No | Class Name | Description |
| 01 | <Name of class> | <Brief description about class ex. One sentence to tell what the class is for, what does it encapsulate> |
|  |  |  |
|  |  |  |

#### External interface

[Describe the external interface of the package (exported classes, methods).]

#### XXX class

[Class description]

* Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | <Attribute name> | <int> |  | <Public/ Static> | <Description of attribute> |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | <method name> | <brief description of method. can be one sentence tell what the method does> |
|  |  |  |
|  |  |  |

* xxxx method

[Method declaration]

[method description, it must be compliance with the brief description in the upper class list]

* Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | in/out | Default | Description |
|  | <parameter name> | <Int> |  |  | <Description of parameter, the special criteria such as boundary should be stated> |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* Implementation

[How to implement the method, it can be in pseudo code or activity diagram or just words]

## Appendix

### Abbreviations and acronyms

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the artifact. This information may be provided by reference to the project's Glossary.]

|  |  |  |
| --- | --- | --- |
| No | Abbreviations and acronyms | Description |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

### References

[This subsection should provide a complete list of all documents referenced elsewhere in the artifact.  Each document is identified by code and name. This information may be provided by reference to an appendix or to another document.]

|  |  |  |
| --- | --- | --- |
| No | Code | Document name |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

## Annexes (Optional)

[Additional material of use to the reader of the Artifact]