

TUTOR ONLINE

DETAIL DESIGN DOCUMENT

Class design

**Project Code: TTO**

**Document Code: TTO-DD – v1.0**

**Hòa Lạc, 9/6/2017**

Record of change

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 9/6/2017 | Create Class Design | A | Create Class Design | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

|  |  |  |
| --- | --- | --- |
| AUTHOR: | Nguyen Thi Khanh Huyen  Team member | 16/06/2017 |
| REVIEWERS: | Tran Viet Vuong  Project Manager (PM) | 16/06/2017 |
| Nong Thi Hoai Thuong  Team member | 16/06/2017 |
| APPROVAL: | Phan Truong Lam  Supervisor | 17/06/2017 |

TABLE OF CONTENTS

1 Introduction 4

1.1 Purpose 4

1.2 Definitions, Acronyms and Abbreviations 4

1.3 References 4

1.4 Overview 4

2 COMMON package and mechanism 4

2.1 Common package 4

2.1.1 Class diagram 4

2.1.2 XXX class 4

2.2 Error, exception handling 4

2.2.1 Class Diagram 4

2.2.2 Usage mechanism 4

2.3 Log, trace and debug 4

2.4 Performance optimizing mechanism 4

2.5 Multilingual processing 4

3 Diagrams 4

3.1 Customer management 4

3.1.1 Add customer 4

3.1.2 Update customer 4

4 Packages 4

4.1 xxx Package 4

4.1.1 Class diagram 4

4.1.2 External interface 4

4.1.3 XXX class 4

4.2 ...... 4

5 Other considerations 4

6 Appendix 4

# Introduction

## Purpose

Tutor Online class design document contains descriptions of all classes of the Tutor Online project. It includes: Class diagrams that describe the static relation of all classes in the system.

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviations | Description | Comment |
| TTO | Tutor Online |  |
| FAQ | Frequently Asked Question |  |

## References

|  |  |
| --- | --- |
| Document Number | Title |
| 1 | Data design document |
| 2 | Software Requirement Specification |

# 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 | int |  |  | <Description of parameter, special criteria such as boundary should be stated> |
|  |  |  |  |  |  |
|  | <return > |  |  |  |  |

##### 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 |
| 01 | <parameter name> | int |  |  | <Description of parameter, the special criteria such as boundary should be stated> |
|  |  |  |  |  |  |
|  | <return > |  |  |  |  |

##### Implementation

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

## ......

# Other considerations

<[This section provides a description of other design elements that were considered as alternatives in selection process for the above class design, i.e. a brief explanation of advantages and disadvantages of the selected package relationships and/or class implementation in comparison with others. It should be a clear answer to the question why the above class design is selected for this system, not the others.>

# Appendix